

Adaptor 全网通

白皮书

文件编号：BS-TWP-百胜 Adaptor 全网通

上海百胜软件股份有限公司

版本修改历史

文档标识:		当前版本:	1.0		
当前状态:	草稿		发布日期:	2023-10-29	
	发布	✓			
修改历史					
日期	版本	作者	修改内容	评审号	变更控制号
2023-09-01	V1.0	吴效银	新建		1.0

目录

一、 产品篇.....	5
1.1. 产品概述.....	5
1.1.1. 品牌客户常见挑战和痛点.....	5
1.1.2. 产品方案.....	5
1.1.3. 产品矩阵.....	6
1.1.4. 产品功能.....	6
1.1.5. 产品价值.....	8
1.2. 预期的读者和阅读建议.....	8
1.3. 名词解释.....	8
二、 技术篇.....	9
2.1. 架构设计.....	9
2.1.1. 系统架构图.....	9
2.1.2. 技术架构图.....	9
2.1.3. 部署架构图.....	10
2.2. 技术特点.....	11
2.2.1. 高性能.....	11
2.2.2. 安全性.....	17
2.3. 开放性.....	22
2.3.1. 百胜开放平台.....	22
2.3.2. 客制化开发.....	22
2.3.3. 商城自助接入 API.....	33
2.3.4. SDK 下载.....	39
2.4. 平台接口调用优化.....	39
2.4.1. 抑制策略说明.....	39
2.4.2. 抑制策略场景.....	39
2.4.3. 抑制策略关键字.....	40
2.4.4. 抑制策略功能说明.....	43
2.5. 编码规范手册.....	45
2.5.1. 编程规约.....	46
2.5.2. 异常日志.....	61
2.5.3. 单元测试.....	64
2.5.4. 安全规约.....	66
2.5.5. MySQL 数据库.....	66
三、 运营篇.....	70
3.1. Adaptor 产品订购及授权开通流程.....	70
3.1.1. 订购地址.....	71
3.1.2. Adaptor 授权及账号开通.....	71
3.2. 计费模式.....	72
3.3. 平台对接需求接入流程.....	72
四、 交付篇.....	73

4.1. Adaptor 配置	73
4.1.1. 标准接口平台	73
4.1.2. 配置说明	73
4.2. 中台/ERP 接入	84
4.2.1. E3+企业中台	84
4.2.2. E3 全渠道中台	91
4.2.3. ME3 全渠道云中台	93
4.2.4. 企业自建中台/ERP	95
4.3. 商城/小程序接入	96
4.3.1. 客制化开发	96
4.3.2. 商城自助接入 API	96
五、 消息服务	96
5.1. 架构设计	97
5.2. 交互说明	97
5.3. 对接平台	98
5.4. Adaptor 平台消息订阅配置	99
5.5. Adaptor 平台接口路由	100
5.6. 平台后台配置消息推送地址	100
5.6.1. 微信视频号小店	100
5.6.2. 喵街	102
5.7. 消息服务管理功能	104
5.7.1. 消息服务 appId 管理	104
5.7.2. 目标服务信息	104
5.7.3. 店铺对照管理	105
5.7.4. 店铺消息推送 topic 订阅管理	105
5.7.5. 消息推送日志	106
六、 常见问题集	107
6.1. 订单通	107
6.2. 打印通	111
七、 附录	114
7.1. SDK 下载	114
7.2. Adaptor 全网通配置文件加解密工具下载	114

一、产品篇

1.1. 产品概述

1.1.1. 品牌客户常见挑战和痛点

- 1、品牌客户在业务发展过程中，不断的拓展新的电商平台，每新增一个平台都需要和平台做一次对接，重新熟悉平台的规则；
- 2、平台用户信息加密改造后怎么触达用户？
- 3、门店 O2O 场景/仓库发货怎么提高时效？
- 4、线上线下会员怎么融合？
- 5、商品怎么快速全渠道铺货？
- 6、平台大促期间怎么保证系统的稳定性？
- 7、是否有安全漏洞问题？
- 8、服务器硬件资源升配置流程繁琐？

1.1.2. 产品方案



- 电商平台的数量一直在不断的增加，每个平台的接口更新频率也在变的更加频繁，品牌客户所需要投入的人力和时间资源非常多，成本高，维护难；
- 在电商平台层和业务系统中间加入 Adaptor 组件，接口统一适配，由 Adaptor 来承载与电商平台的交互工作，对接不同的业务场景，把所有平台数据，统一成标准模型，组件化的拆封压力承载点，提升系统抗压能力；
- Adaptor 作为中间件产品下游可对接不同的业务系统，包括客户自研中台或者其他三方系统，一次对接即可订阅多平台、多场景的应用；

- 1、adaptor 统一接口适配，支持对接上游国内、跨境电商平台；
- 2、下游系统（自研中台，第三方中台等）统一对接 adaptor 标准接口，屏蔽平台的规则变化，支撑品牌业务快速发展和创新；
- 3、支持多个不同的业务场景，包含订单拉取履约发货，面单打印，商品上下架，库存同步，售后处理，短信发货等功能；

4、saas 化部署省去运维成本，按需订阅，即开即用；

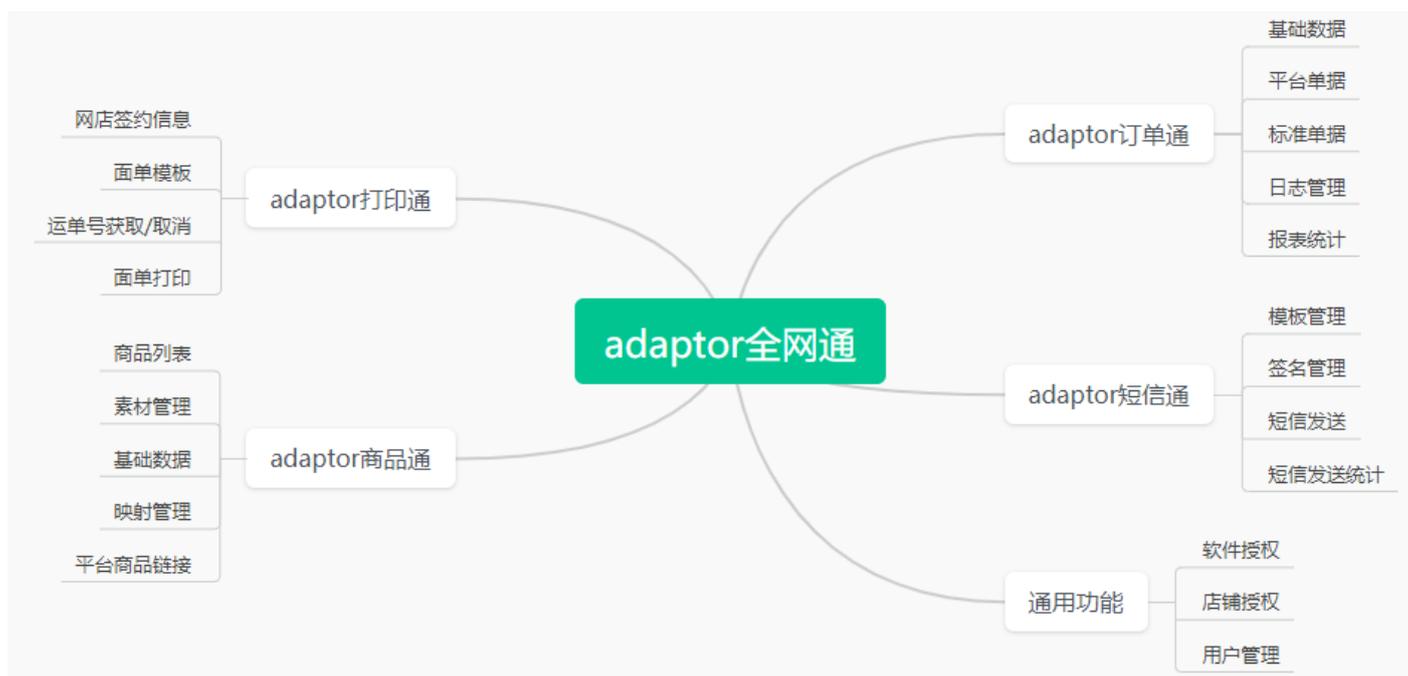
1.1.3. 产品矩阵

adaptor 全网通定位于品牌零售智能导航的互联通道，在不同的业务细分领域满足客户不同的业务诉求，由此独立出单独的细分产品针对性解决问题，从而形成一揽子解决方案，目前有订单通，打印通，短信通，商品通，支付通，发票通，会员通七款产品，未来根据客户需求可以拓展到更多的场景；



1.1.4. 产品功能

adaptor 全网通有基础的公共功能，包括账号体系，软件授权，店铺授权等，在此基础之上，根据不同产品需要解决的问题，设计出不同的功能，主要有：



1.1.4.1. adaptor 订单通

- 1、平台单据拉取，包含订单，售后，发票，商品等，保存原始数据，便于排查问题；
- 2、标准单据：定义标准的单据字段，统一各平台的差异，包含订单，售后，发票，商品等；
- 3、日志管理：对用户操作的日志，接口调用的日志等进行记录，便于在系统异常时查找分析问题；
- 4、报表统计：对于各平台各接口每天的调用次数进行统计，及时发现异常调用，保证接口调用成功率。

1.1.4.2. adaptor 打印通

- 1、网点签约信息，比如使用的是哪家快递，哪个网店，发货地址在哪里；
- 2、模板信息：从平台拉取常用的标准模板和自定义模板，用于面单打印；
- 3、运单号获取/取消：从平台获取运单号，或者取消运单号；
- 4、面单打印：支持生成可打印的图片或者 PDF 文件进行打印。

1.1.4.3. adaptor 短信通

- 1、对短信发送的签名和模板进行管理；
- 2、调用平台短信接口进行短信发送；
- 3、记录每一次短信发送的结果，并且每日统计短信发送情况，便于后续分析对账等。

1.1.4.4. adaptor 商品通

- 1、新建产品资料，对产品的基础资料，属性，图片，价格，库存，详情描述等进行管理；
- 2、素材管理：对商品图片，视频等进行统一管理；
- 3、商品链接：通过生成不同平台的商品链接，可以实现同一款商品在不同平台的差异化管理。

1.1.5. 产品价值



1.2. 预期的读者和阅读建议

本系统设计文档详细阐述了 Adaptor 全网通产品，从产品、技术、运营、交付、运维等方面进行说明，便于相关人员对系统更好地了解、使用、维护和管理。

1.3. 名词解释

- Adaptor 全网通 产品名称
- 订单通 汇聚电商平台单据，并格式化为标准的数据仓库，为下游中台/OMS/ERP 提供统一的数据格式的一揽子解决方案
- 打印通 针对平台加密方案进行面单打印的一揽子解决方案

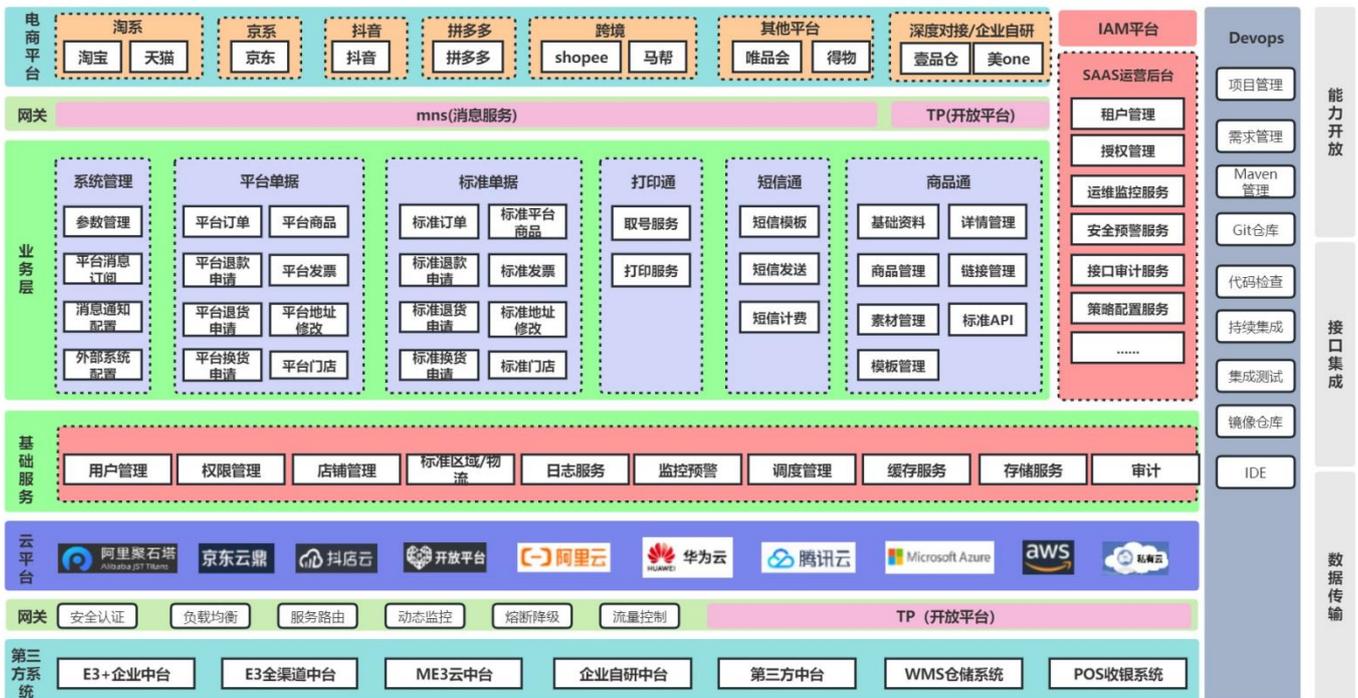
- 短信通 针对平台解密方案进行短信发送的一揽子解决方案
- 商品通 针对商家把商品上架到不同的电商平台进行售卖的一揽子解决方案

二、 技术篇

2.1. 架构设计

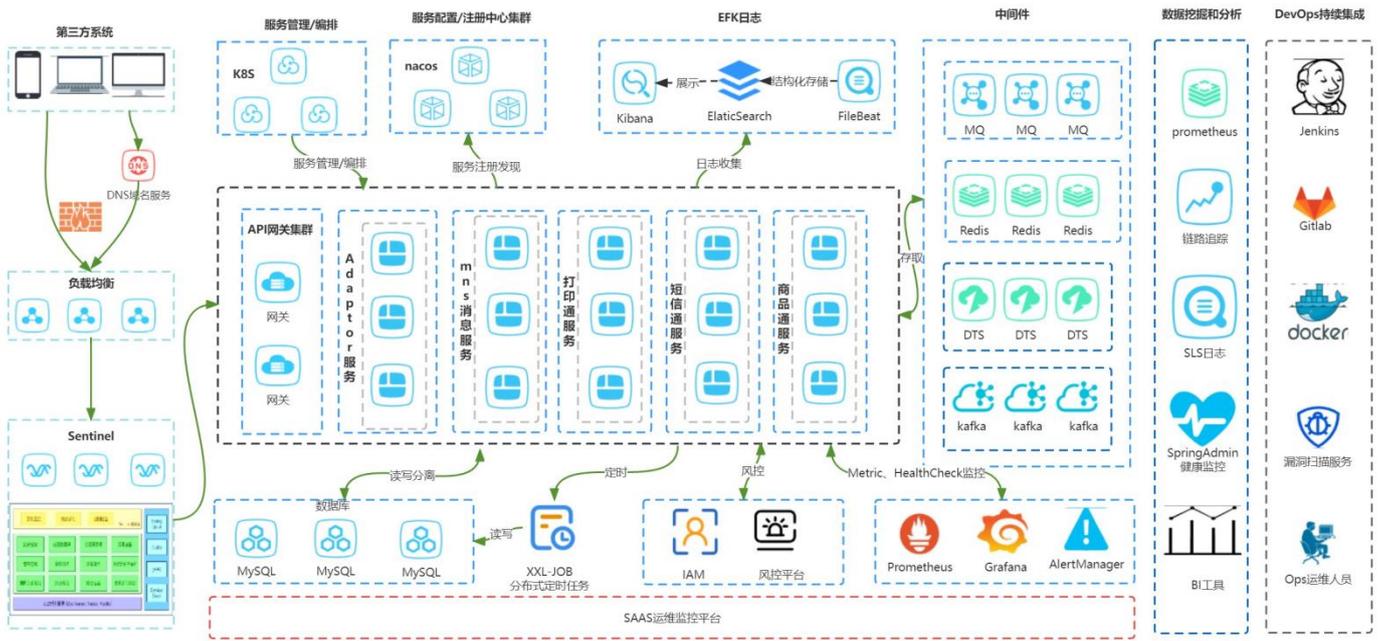
2.1.1. 系统架构图

系统架构图抽象的表示 Adaptor 全网通整体系统框架及各个模块间的相互关系。



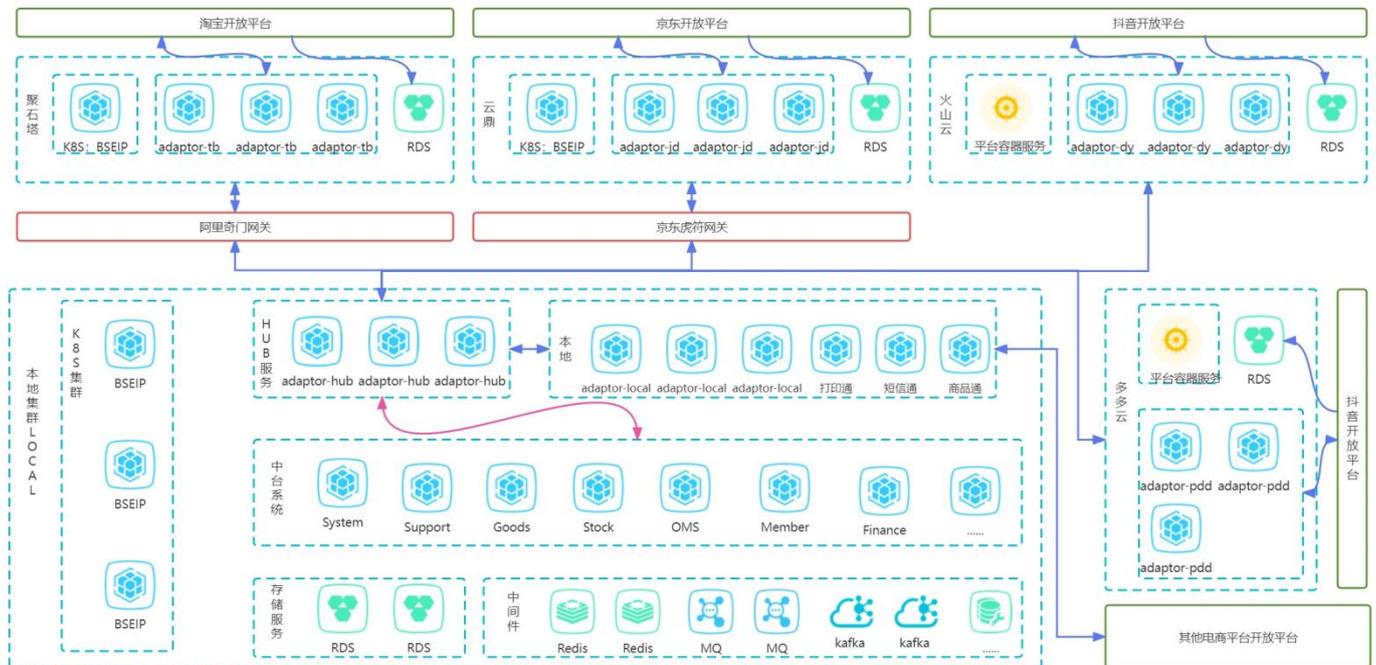
2.1.2. 技术架构图

技术架构图详细列举了 Adaptor 全网通从前端用户进入、负载均衡、网关限流控制、服务编排、服务集群、中间件软件、监控预警、DevOps 持续集成方面所用的技术栈。



2.1.3. 部署架构图

部署架构图详细列举了 Adaptor 全网通涉及的平台、网关、服务、数据库及 Redis、RabbitMQ 等中间件之间的关系。



2.2. 技术特点

2.2.1. 高性能

高并发高性能是现在互联网分布式框架设计必须要考虑的因素之一，它是可以保证系统能同时并发处理很多请求。

Adaptor 是微服务架构的，是基于 Spring Cloud 框架的，Spring Cloud 是 Spring 家族的微服务框架，它提供了一套完整的微服务解决方案。Spring Cloud 提供了一系列的组件，如服务发现、路由、负载均衡、容错处理、配置中心、消息总线等。Spring Cloud 的优点是对 Spring 框架的深度集成和简化了微服务的开发，同时也具有很好的生态系统。

2.2.1.1. 重要指标

1、响应时间

响应时间是指系统或服务从接收请求到作出响应的时间间隔。它通常用于衡量系统或服务的性能。响应时间是完成一次请求所需的总时间，包括处理请求的时间以及在此期间发生的任何等待时间。这个指标可以用来评估系统是否能按照用户的期望快速地响应请求，并且帮助确定是否需要优化系统的性能。较短的响应时间通常被认为是更好的性能指标，特别是对于要求快速响应的实时应用程序或在线服务来说。

2、并发用户数

并发用户数是指在同一时间段内同时使用系统或服务的用户数量。它是衡量系统在处理并发负载时能够支持的用户量的指标。并发用户数可以反映系统的性能、可扩展性和稳定性。

通常，系统或服务会设定一个最大并发用户数的限制，以确保系统能够正常运行而不超过其设计和资源的负载能力。超过最大并发用户数可能导致系统性能下降、响应时间延长或甚至系统崩溃。

通过测量并监控并发用户数，可以评估系统的性能水平、调整系统的资源分配、优化系统的性能以及规划系统的扩展性。这对于拥有高流量、高并发负载的系统非常重要。

3、吞吐量

吞吐量是指系统或服务在单位时间内能够处理的请求数量或事务数量。它是衡量系统性能的一个重要指标，用于评估系统的处理能力和效率。

吞吐量通常以每秒钟处理的请求数量或事务数量来衡量，例如每秒查询率（Queries Per Second, QPS）或每秒请求数（Requests Per Second, RPS）或每秒事务数（Transactions Per Second, TPS）等。

较高的吞吐量表示系统能够处理更多的请求或执行更多的事务，而较低的吞吐量意味着系统处理能力可能受限制。

吞吐量的高低取决于多个因素，包括系统的硬件资源、网络带宽、处理逻辑、数据库性能等。通过优化系统架构、使用高效的算法和数据结构、并发处理等技术，可以提高系统的吞吐量。

吞吐量的关注点通常是在保持系统稳定性和性能的前提下最大化处理能力，尽可能地满足用户的需求并保持良好的用户体验。

Adaptor 主要通过 Redis、RabbitMQ、多线程等提升系统的高并发。

2.2.1.2. 负载均衡

高性能首选方案就是集群化部署，一台服务器承载的 QPS 有限，多台服务器叠加效果就会有明显提升。集群化部署，就需要考虑如何将流量转发到服务器集群，这里就需要用到负载均衡。

Adaptor 基于平台云负载均衡能力来实现集群的负载均衡，比如阿里云集群使用的 SLB、华为云集群使用的 ELB 等。

2.2.1.3. 池化技术

复用单个连接无法承载高并发，如果每次请求都新建连接、关闭连接，考虑到 TCP 的三次握手、四次挥手，需要花费大量开销。

池化技术的核心是资源的“预分配”和“循环利用”，常用的池化技术有线程池、连接池、进程池、对象池、内存池、协程池。连接池的几个重要参数：最小连接数、空闲连接数、最大连接数。

Adaptor 使用线程池、数据库连接池技术来进行资源的复用，降低系统消耗，提升系统吞吐能力。

2.2.1.4. 风控拦截

互联网高并发流量并不都是纯净的，也有很多恶意流量（比如黑客攻击、恶意爬虫、黄牛、秒杀器等），我们需要设计流量拦截器，将哪些非法的、无资格的、低优先级的流量过滤掉（风控掉），减轻系统的并发压力，将核心能力留给正常业务请求。

Adaptor 接入了百胜 WAF（Web Application Firewall，Web 应用防火墙），借助 WAF 能力对请求异常进行检测，过滤非法流量；接入 Sentinel 网关，Sentinel 是面向分布式服务架构的高可用流量防护组件，主要以流量为切入点，从限流、流量整形、熔断降级、系统负载保护、热点防护等多个维度来帮助开发者保障微服务的稳定性。

2.2.1.5. 缓存技术

MySQL 即使使用分库分表，读写分离，完美的连接池配置等也无法抵挡 QPS 大于 10W 带来的冲击。Adaptor 通过使用内存缓存，缓存预热读多级缓存（JVM 缓存，其次 Redis），减少对 MySQL 的压力。

2.2.1.6. 多线程

在微服务架构中，多线程是一种常用的技术手段，用于处理并发请求和提高系统的性能和吞吐量。多线程允许同时执行多个任务，从而实现并发处理。

以下是一些常见的使用多线程的情况和技术：

- 1、请求并发处理：微服务架构中，服务可能会同时收到多个请求。通过使用多线程，可以并发地处理这些请求，提高系统的吞吐量。每个请求可以在一个独立的线程中执行，从而充分利用服务器的处理能力。
- 2、异步操作：微服务中有些操作可能是耗时的，例如从数据库读取数据或进行网络请求。通过使用多线程，可以将这些耗时的操作放在独立的线程中执行，而不会阻塞主线程。这样可以提高系统的响应速度，使服务在等待耗时操作完成时能够处理其他请求。
- 3、并行计算：在某些情况下，微服务需要进行大量的计算或处理复杂的任务。使用多线程可以将这些计算任务分解成多个子任务，并且可以同时多个线程中进行计算。这样可以加快计算速度，提高系统性能。
- 4、线程池：为了更好地管理线程资源和控制并发度，通常使用线程池来创建和管理线程。线程池维护一组可重用的线程，可以通过配置线程池的大小来控制并发度。每当有新的任务到达时，线程池会从池中获取一个可用线程来执行任务，任务执行完成后将线程返回给线程池。

需要注意的是，在多线程编程中需要注意线程安全性和竞态条件的问题。合理地进行线程同步和锁定，以避免不正确的结果和并发问题。同时，对于高并发的场景，也需要进行性能测试和调优，以确保多线程的使用能够带来实际的性能提升。

2.2.1.7. 优化 IO

网络请求在每次建立连接，数据交互，释放连接都会消耗大量的资源，同时涉及到用户态到核心态的切换，耗费了大量的 IO 时间。

Adaptor 通过将前端的多次单个的请求，优化为一次批量请求，减少网络 IO，提升响应。对于 MySQL 处理场景，通过批量插入，批量查询的方式，减少网络 IO。

2.2.1.8. 优雅打印日志

在高并发情况下，日志打印不但会占用程序的 IO，增加响应耗时，还可能因为日志量过大，导致磁盘打满，从而导致整个集群挂掉，所以对于控制应用程序的日志打印非常重要。

Adaptor 在生产环境只打印 error 级别的日志不打印 info 的日志，减少应用的日志量；并且应用支持按照模块的方式配置日志打印级别，如果遇到某一个模块需要 info 日志进行问题定位的，可以优雅的短时间内打印 info 日志，提升问题排查的能力。

2.2.1.9. 横向扩展

横向扩展是指通过向系统添加更多的实例或节点的方式来增加系统的处理能力和容量。也被称为水平扩展或扩容。在横向扩展中，系统的负载被均匀地分散到多个相同或相似的实例或节点上，每个实例或节点负责处理一部分请求。这样可以提高系统的并发性能、吞吐量和可伸缩性，并且不会对现有的单个实例产生过多的负担。当负载增加时，可以简单地通过添加更多的实例或节点来实现横向扩展，从而满足不断增长的需求。

横向扩展的优势包括：

- 1、提高系统的并发处理能力和吞吐量，能够处理更多的请求或事务。
- 2、支持更高的负载和流量，保持系统的响应性能和稳定性。
- 3、具备良好的可伸缩性，可以根据需求动态地增加或减少实例数量。
- 4、提高系统的可用性，单个实例或节点的故障对整体系统影响较小。
- 5、更灵活和经济高效，不需要过多的硬件资源投入。

横向扩展在云计算、大数据处理、分布式系统和微服务架构等场景中得到广泛应用，有助于构建高性能、可扩展和弹性可靠的系统。

Adaptor 基于微服务架构，将一个大型应用拆分成多个小而独立的服务，每个服务可以独立部署和扩展。

Adaptor 集群可以按需扩展，可以通过添加更多的服务实例来满足高负载的需求。

2.2.1.10. 异步通信

在传统的微服务架构中，服务之间的通信通常是同步的。即当一个服务发起请求时，它必须等待其他服务的响应才能继续执行。这种方式在一些场景下会导致性能瓶颈，影响整体的系统响应时间。

而异步通信则是指调用方在发送请求后可以继续执行后续操作，而无需等待接收方的回复。接收方将请求接收后，通过消息中间件异步处理后再回复调用方。在微服务架构中，由于服务间调用非常频繁，如果全部采用同步通信方式，则会造成大量阻塞，影响系统性能。因此，使用异步通信方式能够更好地解决这个问题。异步通信方式可以给系统带来以下的一些提升：

- 1、降低服务间的耦合性：通过使用消息代理，服务之间的通信变得解耦，每个服务都可以独立处理收到的消息。
- 2、提高系统的可伸缩性：异步通信允许服务在处理请求时能够同时处理其他请求，从而提高了系统的吞吐量。
- 3、增加系统的弹性和容错性：通过消息队列，我们可以实现消息的持久化和重试机制，从而增加了系统对错误和故障的容错能力。

通过异步通信确实是能进行系统解耦，提升系统的吞吐，但是如何保证异步消息都能被正确的执行，不会有异步消息被无缘无故的丢弃等，这里就要提到两个原则

原则一：幂等

在微服务架构中，幂等性是一个非常重要的原则，即相同的参数无论执行多少次，无论调用相同的 API 多少次，对后端产生的影响是一致的，结果都是相同的，并且不会有额外的副作用。但是这里有一个容易被误解的地方，多次执行、调用结果一致，并不是说每次执行、调用后端返回的信息都是一致的，幂等强调的是对后端数据产生的影响，而不应该仅仅关注在返回的信息。

原则二：最终一致性

CAP 原则又称 CAP 定理，指的是在一个分布式系统中，一致性（Consistency）、可用性（Availability）、分区容错性（Partition tolerance）。CAP 原则指的是，这三个要素最多只能同时实现两点，不可能三者兼顾。基于这个原则，在微服务设计的时候需要做出一定的取舍：如果舍弃 P 则意味着放弃了系统的可扩展性，不能扩展子节点，这就违背了分布式系统设计的初衷；如果舍弃 A 则意味着一旦发生了网络故障或者消息丢失等请求的时候就必须牺牲掉用户的体验，等待所有数据全部一致了之后再让用户访问系统，这会用户体验非常不友好，消磨用户耐心，进而可能导致失去用户；如果舍弃了 C，则意味着分布式系统数据存在差异，不一致，从而导致业务流程的错误，看起来也是不能舍弃的。

BASE 理论，BASE 是 Basically Available（基本可用）、Soft state（软状态）和 Eventually consistent（最终一致性）三个短语的简写，BASE 是对 CAP 中一致性和可用性权衡的结果，其来源于对大规模互联网系统分布式实践的结论，是基于 CAP 定理逐步演化而来的，其核心思想是即使无法做到强一致性（Strong consistency），但每个应用都可以根据自身的业务特点，采用适当的方式来使系统达到最终一致性（Eventual consistency）。

Adaptor 在单据格式化之后推送中台不使用同步推送的方式，通过把需要推送的单据给到 RabbitMQ，由 RabbitMQ 进行异步调度，通过幂等、最终一致性原则，保证单据的最终一致性，从而来提升系统的吞吐量。

2.2.1.11. 精细化调优

微服务中的精细化调度是指对微服务的部署和调度进行细致和精确的控制，以实现更高效、灵活和可靠的运行。通过精细化调度，可以根据实际需求和资源状况，动态地分配和调度微服务实例的位置和数量。

精细化调度通常涉及以下方面：

- 1、资源分配：根据微服务的需求和系统的资源情况，精确地分配计算资源（如 CPU、内存、磁盘）和网络带宽，以满足微服务的运行要求。
- 2、弹性伸缩：根据负载的变化实时调整微服务实例的数量，以确保系统在高峰期能够扩展以处理更多的请求，并在低峰期自动缩减以节省资源。
- 3、故障转移和负载均衡：通过监控微服务的状态和性能指标，及时发现故障实例并进行恢复与替换。通过负载均衡技术将请求动态分发到可用的实例上，避免单个实例过载。
- 4、策略调度：根据具体业务和需求，制定适合的调度策略。例如，可以实现基于优先级、权重、成本等因素的调度方式。
- 5、低延迟调度：通过将微服务部署在靠近用户或客户端的节点上，减少网络延迟，提高响应速度和性能。

实现微服务的精细化调度可以提高系统的资源利用率，优化服务的性能和可靠性，以及提供更好的用户体验。这需要依赖于可扩展的部署平台（如容器化平台或自动化工具）和调度策略的智能化调控。

Adaptor 中每个微服务都是独立部署的，开发团队可以对每个服务进行更细粒度的性能调优。基于 K8S 进行部署，可以实现服务的弹性伸缩及故障转移等。Adaptor 使用 XXL-JOB 管理调度，可以根据调度的不同分类配置不同的运行策略以及对应的服务器资源，资源使用达到最优，从而提高系统的整体性能。

2.2.1.12. 容错和恢复能力

在微服务架构中，各个微服务通常都运行在独立的容器中，并相互依赖以提供完整的应用程序功能。这种分布式架构的优势在于每个微服务可以独立开发、测试和部署，但同时也带来了一些挑战。以下是一些常见的微服务故障类型：

- 1、服务不可用：某个微服务由于各种原因（例如硬件故障、网络问题或软件错误）而无法提供服务。
- 2、响应时间延迟：微服务的响应时间增加，导致整个应用程序的性能下降。
- 3、异常和错误：微服务可能在处理请求时抛出异常或返回错误响应。
- 4、资源耗尽：微服务可能消耗了所有可用的资源，如内存、CPU 或数据库连接。

为了应对这些故障，需要采取适当的故障恢复和容错策略，有以下常用的策略：

1、超时设置

在微服务之间进行通信时，始终要设置合理的超时时间。如果一个微服务无法在指定的时间内响应请求，您应该考虑将其标记为不可用，并尝试其他备用服务。

2、断路器模式

断路器模式是一种故障恢复策略，它可以防止故障的微服务对整个系统的影响。当微服务的错误率达到一定阈值时，断路器会打开，将请求快速失败，而不是等待响应超时。Hystrix 是一个流行的断路器库，它可以轻松地集成到 Spring Cloud 中。

3、负载均衡和多副本部署

通过在微服务之间实现负载均衡，可以将流量分散到多个副本或实例中。如果某个实例发生故障，负载均衡器会自动将流量路由到健康的实例上。

4、重试机制

在发生故障时，重试请求可能有助于恢复服务。您可以使用 Spring Retry 等库来实现请求重试。

5、日志和监控

记录和监控是故障排除的关键工具。确保微服务应用程序生成足够的日志信息，以便在出现问题时进行故障排查。使用监控工具来实时监视应用程序的性能和可用性。

6、服务降级

当发生故障时，有时最好提供有限但可用的服务，而不是完全失败。服务降级策略允许定义在故障情况下提供的最小功能。

Adaptor 通过使用服务注册与发现机制和负载均衡器来实现容错和恢复能力。当某个服务发生故障或不可用时，系统可以自动将请求路由到可用的服务实例上，从而保持系统的可用性和性能。

2.2.1.13. 运维效率

微服务通过以下方式提升运维效率

- 1、解耦和独立部署：微服务架构将应用程序拆分为一组小型、独立的服务。这使得每个服务都可以独立部署、升级和扩展，而不会对其他服务产生影响。这种解耦减少了运维的复杂性，可以更快速地进行部署和更新，

同时降低了故障的风险。

- 2、自动化运维：微服务架构鼓励使用自动化工具和技术来管理和监控服务。通过自动化，运维团队可以大大减少手动操作，例如自动化部署、自动化监控、自动化日志收集和分析等。这种自动化可以提高效率，减少人为错误，并实现快速响应和故障排除。
- 3、弹性扩展：微服务架构的一个重要优势是可以针对每个服务单独进行弹性扩展。通过根据需求调整每个服务的实例数量，可以更好地处理高负载情况，并在低负载时减少资源使用。这有效地提高了系统的运维效率，使得资源得到更好地利用。
- 4、面向团队的组织结构：微服务架构通常倡导团队的自治性和负责制。每个微服务都由一个独立的开发团队负责，并且负责运维和监控自己的服务。这种面向团队的组织结构可以提高团队的效率和自主性，因为每个团队可以专注于自己负责的服务，减少了沟通和决策的复杂性。
- 5、容器化和编排技术：微服务架构通常与容器化技术（如 Docker）和容器编排技术（如 Kubernetes）结合使用。通过将每个微服务封装到一个独立的容器中，并使用容器编排工具进行管理，可以实现更高效的部署、扩展和管理。容器化还可以提供更好的隔离性和资源利用率。

综上所述，微服务通过解耦和独立部署、自动化运维、弹性扩展、面向团队的组织结构以及容器化和编排技术等手段，提升了运维效率，使得系统更加可靠、弹性和易于管理。

Adaptor 基于百胜 DevOps 体系，来实现运维的自动化、可视化，提升运维的效率。

2.2.2. 安全性

2.2.2.1. 客户端安全性

1、XSS 跨站脚本攻击

对所有请求参数都进行规则过滤。对其中有恶意脚本提示非法请求，停止后续请求，从而达到用户的恶意攻击的目的。

2、CSRF 跨站请求伪造

通过 HTTPReferer 验证跨域请求来源的合法性。

2.2.2.2. 输入输出安全性

1、SQL 注入攻击

对所有请求参数都进行 SQL 规则过滤，包含非法字符串的请求停止执行，防止服务器执行恶意的 SQL 命令。

2、表单篡改

前后端约定对表单数据进行签名 Sign，后端对表单签名进行校验，防止表单篡改。

2.2.2.3. 认证安全性

1、身份认证安全

使用 HTTPS 通道，是以安全为目标的 HTTP 通道，在 HTTP 的基础上通过传输加密和身份认证保证了传输过程的安全性；接入百胜 IAM，对账号风险进行管控；接入百胜 WAF；

权限控制基于 OAuth 2.0 授权框架，用于委托访问资源的权限。在 Adaptor 中，使用 OAuth 2.0 来管理令牌的生成、验证和刷新。通过 OAuth 2.0，可以实现单点登录、授权码、密码、客户端凭证等多种认证方式。

2、登录暴力破解

- a) 接入百胜 IAM；
- b) 第一次登录失败之后强制输入验证码；
- c) 连续 5 次登录失败的，账号强制锁定，需要连续管理员进行解锁才能再次登录；
- d) 系统提供 MD5+Salt 方式存储密码；
- e) 登录之后超过 2 个小时未操作，强制退出

3、系统弱密码

密码必须包含大写字母、小写字母、数字、特殊符号，且长度不能少于 8 位；

4、密码强制定期修改

超过 60 天密码强制失效，登录时提示“密码已过期，请重新修改密码”；

2.2.2.4. 权限控制安全性

1、权限控制

控制对应请求地址的权限控制，无权限页面限制提示无权限，并对后台接口请求增加访问控制；

2、越权访问-垂直权限访问

基于 RBAC 的权限控制，根据角色管控模块菜单及页面操作按钮是否可见；

3、越权访问-水平权限访问

可对平台权限进行控制，管控在标准单据、平台单据中是否可查看对应平台数据；

4、未授权访问

未授权访问的目录，提示无权限；

5、目录列表

在任何情况下均不允许遍历目录。

2.2.2.5. 会话管理安全性

1、Cookie 加 httponly 属性

通过限制 web 页面程序的 browser 端 script 程序读取 cookie，保障了 Cookie 信息的安全，不被外部恶意读取盗用；

2、Cookie 加 secure 属性

在启用 HTTPS 的场景下，浏览器仅通过 HTTPS 连接传回 cookie，确保 cookieID 是安全的；

3、会话标识未更新

控制每次用户登出都会对会话进行失效处理，登陆重新生成 token；

4、会话清除

在用户退出时，立即清除 token，防止他人盗用；

5、用户凭证失效

通过参数设置用户凭证过期时间，默认 2 小时。

2.2.2.6. 数据安全性

1、敏感数据展现

涉及订单敏感信息、用户敏感信息的数据系统界面脱敏展示，例如用户手机号、用户收货地址；

2、敏感数据传输

涉及敏感信息的数据，系统默认以密文传输；

3、敏感数据存储

敏感数据在数据库中保持加密存储；

4、服务器文件安全

在任何情况下不允许遍历目录；

5、log 日志文件安全

log 日志文件不允许下载。

2.2.2.7. 文件安全性

1、文件上传

- a) 文件后缀、文件大小限制；
- b) 上传目录去除可执行权限；
- c) 通过文件流生成文件唯一标识，提交后台验证，校验是否为合法的文件类型；

2、文件下载

通过重定向对文件下载类型及权限的控制，防止恶意下载。

2.2.2.8. 配置安全性

1、系统日志清理

系统提供日志清除策略，根据策略自动清除日志；

2、系统安全日志

系统提供登陆、订单链路节点操作等日志记录。

2.2.2.9. 接口安全

1、拒绝服务攻击

通过接入 gateway 网关，来进行拒绝服务攻击监测、拦截。

2.2.2.10. 敏感信息泄露

1、通讯加密

使用 HTTPS 的方式进行交互，并接入到百胜 WAF；前后端在交互的时候，对表单数据均进行签名，防止对表单数据篡改

2、中间件配置安全

RabbitMQ、Redis 等账号信息均使用加密方式配置，有效降低外部获取此类账号信息之后的损失；RDS 的密码配置也是通过加密方式配置的。

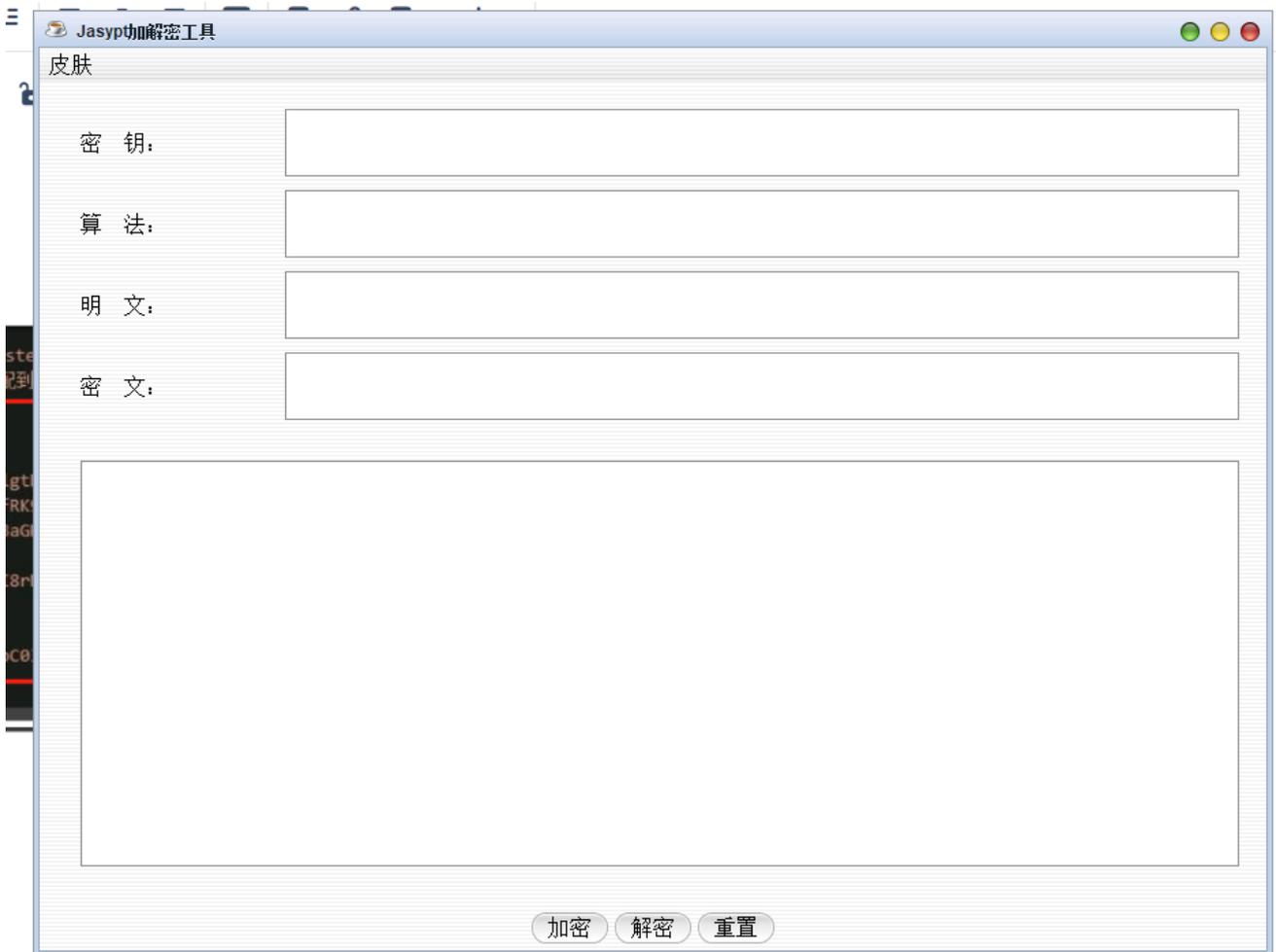
在通过加密 jar 工具加密后的的效果如下，在维护到 nacos 中的时候，前缀添加 ENC(.)

```

23   dynamic:
24     primary: master #设置默认的数据源或者数据源组,默认值即为master
25     strict: false #设置严格模式,默认false不启动。启动后在未匹配到指定数据源时候会抛出异常,不启动则使用默认数
26   datasource:
27     master_0:
28       url: ENC(UG6BZZX33Ax
+p/gL1hLEytedIqtVNIbBaAI8ngp1PyDUIN7zQPAB16qKsrnXqryEhzKxSI1gtNx8VARNFQ6TI0brorc0h7mQH9eIG9an4XLLWc7yYI
+wQs8ASnurDxIzWqTLu1k0Vc88wD50W3GXmTEbUbNKRJA15S7TePcj/aK+bgfRK9zAThzzNfTSC
+By2JLY8ZzBy1U5hrPFZyeYB5ZBc5As/Ufr/i0o8TXsIKP0heLA+XArz1dp/BaGH1gA/03oqe0nyXmLxBRyqX
+sChirMSDrBC9XCcx6no1zEzK637icnPjzlxUqY5+pEAM4qxJDH2r57gOBDx
+P46eGT0k1iPcLVi9dqchChGAe3sEeZ5bveH8HPmWp/V361FkwEyIkWyy5VVI8rNNOzfTalZ6pTyX4S8ungSxbDva/H63GdwCFdXAckI
/zU5AwNBZAdM1EDECfGg/+L0rht3H4Q=)
29       username: ENC(Bzft1rY/PzXzG/x/TJ+c22YBMrq1xXoG)
30       password: ENC(4fM0UvbB+Q1H51b12M70XjcMZsu5dqjjs0oJpC03o/bhNZ0meEqdvQ==)
31       driver_class_name: com.mysql.cj.jdbc.Driver
32     hikari:

```

加密工具运行截图:



加密工具中需要的密钥、算法等，配置在 adaptor 工程下的 bootstarp.yml 中



adaptor 全网通配置文件加解密工具下载：详见附录

2.3. 开放性

2.3.1. 百胜开放平台

Adaptor 通过百胜开放平台对外提供开放服务，百胜开放平台地址：<https://open.baison.net/>，在百胜开放平台常用工具部分，有 Adaptor 接口对接指南，及 API 调用测试工具可供测试验证使用。

2.3.2. 客制化开发

Adaptor 把核心底层封装为了 common 包，项目上面如果要客制化开发对接电商平台的，可以通过引入 common 包的方式进行客制化开发。

下面对客制化开发的步骤进行详述。

前置步骤：申请百胜 VPN 账号

申请路径：钉钉 OA 审批->信息中心->非百胜人员账号管理流程

申请通过之后，还需要安装 VPN 软件，连接 VPN 之后即可开始客制化开发。软件可联系信息中心获取。

步骤一：maven 配置百胜私服地址，settings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
```

```
<localRepository>F:\11-programData\Maven</localRepository>
<pluginGroups>
</pluginGroups>
<proxies>
</proxies>
<servers>
  <server>
    <id>e3plus-releases</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
  <server>
    <id>e3plus-snapshots</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
  <server>
    <id>3th_part</id>
    <username>admin</username>
    <password>admin123</password>
  </server>
</servers>
<mirrors>
  <mirror>
    <id>nexus1</id>
    <mirrorOf>central</mirrorOf>
    <url>http://192.168.149.221:8888/nexus/repository/maven-public/</url>
  </mirror>
</mirrors>
```

```
<profiles>
  <profile>
    <id>e3plus</id>
    <repositories>
      <repository>
        <id>e3plus-3rd</id>
        <name>e3plus 3rd</name>
        <url>
          http://192.168.149.221:8888/nexus/repository/third_repo/
        </url>
        <snapshots>
          <enabled>true</enabled>
        </snapshots>
        <releases>
          <enabled>true</enabled>
        </releases>
      </repository>
      <repository>
        <id>e3plus-releases</id>
        <name>e3plus-releases</name>
        <url>
          http://192.168.149.221:8888/nexus/repository/e3plus-releases/
        </url>
        <snapshots>
          <enabled>true</enabled>
        </snapshots>
        <releases>
          <enabled>true</enabled>
        </releases>
    </repositories>
  </profile>
</profiles>
```

```
</repository>
<repository>
  <id>central</id>
  <name>central</name>
  <url>
    http://central
  </url>
  <snapshots>
    <enabled>true</enabled>
  </snapshots>
  <releases>
    <enabled>true</enabled>
  </releases>
</repository>
</repositories>
</profile>
<profile>
  <id>jdk-1.8</id>
  <activation>
    <activeByDefault>true</activeByDefault>
    <jdk>1.8</jdk>
  </activation>
  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>
  </properties>
</profile>
</profiles>
```

```
<activeProfiles>
    <activeProfile>e3plus</activeProfile>
</activeProfiles>
```

</settings>

步骤二：maven 引入 common 包依赖

```
<dependency>
    <groupId>com.baison</groupId>
    <artifactId>adaptor-saas-common-base</artifactId>
    <version>0.0.2-SNAPSHOT</version>
</dependency>
```

步骤三：开发说明

下面以增加 tuzhan（兔展）平台对接为例子

代码结构说明：

类别	位置	备注
Entity	\adaptor-custom-api\src\main\java\com\baison\e3plus\adaptor\api\tuzhan\	model: 实体 service: 接口
Mapper	\adaptor-custom-service\src\main\resources\mapper\tuzhan\	mybatis 资源 Sql 文件
Liquibase	\adaptor-custom-service\src\main\resources\liquibase\release\tuzhan\1.0.1	初始化脚本
Control 层	\adaptor-custom-service\src\main\java\com\baison\e3plus\adaptor\service\tuzhan\controller\	前端入口
Dao 层	\adaptor-custom-service\src\main\java\com\baison\e3plus\adaptor\service\tuzhan\dao\	
Job 入口	\adaptor-custom-service\src\main\java\com\baison\e3plus\adaptor\service\tuzhan\job\api\	调度任务入口
Service 实现层	\adaptor-custom-service\src\main\java\com\baison\e3plus\adaptor\service\tuzhan\service\impl\	

下面以订单场景为例子进行代码开发说明：

场景	文件名称	类	备注
订单实体	/model/ TuzhanTrade.java	<code>public class TuzhanTrade extends AbstractBean {}</code>	
订单接口	/service/ ITuzhanTrade Service.java	<code>public interface ITuzhanTradeService extends IService<TuzhanTrade> {}</code>	
订单列表下载入口（计划任务）	/job/api/ TuzhanPullPlatform TradeJobHandler.java	<code>public class TuzhanPullPlatformTradeJobHandler extends AbstractPullPlatformHandler<PlatformTradeRequestDTO> {}</code>	
订单详情下载入口（计划任务）	/job/api/ TuzhanPullTrade QueueJobHandler.java	<code>public class TuzhanPullTradeQueueJobHandler extends AbstractPullPlatformQueueHandler<TempOriginTradeQueue> {}</code>	
订单格式化入口（计划任务）	/job/api/ TuzhanPullOrigin TradeJobHandler.java	<code>public class TuzhanPullOriginTradeJobHandler extends AbstractPullOriginFormatJobHandler {}</code>	
订单前端操作入口	/controller/ TuzhanTrade Controller.java	<code>public class TuzhanTradeController extends AbstractController {}</code>	
订单处理服务	/service/impl/ TuzhanTrade ServiceImpl.java	<code>public class TuzhanTradeServiceImpl extends ServiceImpl<TuzhanTradeMapper, TuzhanTrade> implements ITuzhanTradeService, IPlatformOriginalMessageService<PlatformTradeOriginalMessageDTO>, IPlatformRequestService<PlatformTradeRequestDTO>, IWriteBackPlatformTradeService {}</code>	
订单处理Dao	/dao/ TuzhanTrade Mapper.java	<code>public interface TuzhanTradeMapper extends BaseMapper<TuzhanTrade> {}</code>	

场景一：订单列表下载场景

TuzhanPullPlatformTradeJobHandler 需要继承 AbstractPullPlatformHandler 基类，并实现、重写基类中的方法实现根据不同的平台进行订单列表下载功能。

//调度任务 code, xxl-job 中处理的 handler

```
@JobHandler(value = "TuzhanPullPlatformTradeJobHandler")
```

```
@Component
```

```
public class TuzhanPullPlatformTradeJobHandler extends AbstractPullPlatformHandler<PlatformTradeRequestDTO> {
```

```
//
```

```
}
```

方法	实现	备注
getPlatform	<pre>@Override public String getPlatform() { return PlatformEnum.TUZHAN.getCode(); }</pre>	获取平台编码
getLastModifiedTimeField	<pre>@Override public SFunction<Shops, ?> getLastModifiedTimeField() { return Shops::getLastTradeSyncTime; }</pre>	获取店铺订单最后下载时间
loadPlatformTrade	<pre>@Override public void loadPlatformTrade(PlatformTradeRequestDTO requestDTO, Shops shops, Date curDate) { // @todo }</pre>	父类中 loadPlatformTrade 方法为统一的列表下载方法包括开始时间、结束时间的入参方式等，如果当前平台下载方式不能覆盖，可重写此基类方法，进行定制化开发
doLoadAndSavePlatformData	<pre>@Override public void doLoadAndSavePlatformData(PlatformTradeRequestD TO requestDTO, Date curDate, String shopCode, Integer curPageNo) { // @todo }</pre>	需要实现父类中的 public abstract void doLoadAndSavePlatformData(ReqDto requestDTO, Date curDate, String code, Integer curPageNo); 方法，实现实际的平台接口请求及返回报文解析功能
getTotalPages	<pre>@Override public Integer getTotalPages(PlatformTradeRequestDTO requestDTO) { // @todo }</pre>	需要实现父类中的 public abstract Integer getTotalPages(ReqDto requestDTO); 方法，实现根据平台返回报文解析并计算出总的分页数

buildQueryParam	<pre> @Override public PlatformTradeRequestDTO buildQueryParam(Shops shops, Date curDate, JSONObject parseObject) { // @todo } </pre>	需要实现父类中的 <code>public abstract ReqDto</code> <code>buildQueryParam(Shops shops, Date curDate, JSONObject parseObject);</code> 方法，实现根据调度任务入参及店铺信息等入参组装平台请求所需要的参数信息
-----------------	---	--

场景二：订单详情下载场景

TuzhanPullTradeQueueJobHandler 需要继承 AbstractPullPlatformQueueHandler 基类，并实现、重写基类中的方法实现根据不同的平台进行订单详情下载功能。

//调度任务 code, xxl-job 中处理的 handler

```
@JobHandler(value = "TuzhanPullTradeQueueJobHandler")
```

```
@Component
```

```
public class TuzhanPullTradeQueueJobHandler extends AbstractPullPlatformQueueHandler<TempOriginTradeQueue> {
```

```
//
```

```
}
```

方法	实现	备注
getPlatform	<pre> @Override public String getPlatform() { return PlatformEnum.TUZHAN.getCode(); } </pre>	获取平台编码
formatKey	<pre> @Override public String formatKey() { return PlatformEnum.TUZHAN + "_" + FormatKeyEnum.FORMAT_TRADE.getCode(); } </pre>	需要实现父类中 <code>public abstract String formatKey();</code> 方法，获取插入到 redis 中的键值
loadPlatformTradeDetail	<pre> @Override public void loadPlatformTradeDetail(TempOriginTradeQueue queues, Shops shops) { } </pre>	需要实现父类中 <code>public abstract void</code> <code>loadPlatformTradeDetail(TempQueue queues, Shops shop);</code> 方法，实现实际的平台详情接口请求及返回报文的解析功能

场景三：订单格式化场景

TuzhanPullOriginTradeJobHandler 需要继承 AbstractPullOriginFormatJobHandler 基类，并实现、重写基类中的方法实现根据不同的平台进行订单格式化功能。

//调度任务 code, xxl-job 中处理的 handler

@JobHandler(value = "TuzhanPullOriginTradeJobHandler")

@Component

```
public class TuzhanPullOriginTradeJobHandler extends AbstractPullOriginFormatJobHandler {
}
```

方法	实现	备注
formatKey	<pre>@Override public String formatKey() { return PlatformEnum.TUZHAN + "_" + FormatKeyEnum.FORMAT_TRADE.getCode(); }</pre>	需要实现父类中 public abstract String formatKey(); 方法，获取插入到 redis 中的键值
doFormat	<pre>@Override public Result<Object> doFormat(List<String> idList) { return iPlatformOriginalMessageService.format(idList); }</pre>	需要实现父类中 public abstract Result<Object> doFormat(List<String> idList); 方法，实现订单实际格式化

订单处理类

TuzhanTradeServiceImpl 需要继承 ServiceImpl 基类，实现 ITuzhanTradeService、IPlatformOriginalMessageService、IPlatformRequestService、IWriteBackPlatformTradeService 接口，实现、重写基类、接口中的方法实现根据不同的平台进行订单处理业务。

@Service

```
public class TuzhanTradeServiceImpl extends ServiceImpl<TuzhanTradeMapper, TuzhanTrade> implements
ITuzhanTradeService, IPlatformOriginalMessageService<PlatformTradeOriginalMessageDTO>,
IPlatformRequestService<PlatformTradeRequestDTO>, IWriteBackPlatformTradeService {
}
```

方法	实现	备注
create	<pre>@Override public boolean create(TuzhanTrade tuzhanTrade) { return save(tuzhanTrade); }</pre>	需要实现父类中的 boolean create(TuzhanTrade tuzhanTrade); 方法，把 trade 进行存储
update	<pre>@Override public boolean update(TuzhanTrade tuzhanTrade) { return updateById(tuzhanTrade); }</pre>	需要实现父类中的 boolean update(TuzhanTrade tuzhanTrade);

		方法，进行订单信息的更新操作
delete	<pre> @Override public boolean delete(Long id) { if (id == null) { logger.error("删除记录 id 不能为空!"); return false; } return removeById(id); } </pre>	需要实现父类中的 <code>boolean delete(Long id);</code> 方法，进行订单信息的删除操作
findById	<pre> @Override public TuzhanTrade findById(Long id) { return tuzhanTradeMapper.selectById(id); } </pre>	需要实现父类中的 <code>TuzhanTrade findById(Long id);</code> 方法，进行订单信息的查询操作
queryPage	<pre> @Override public PageInfo<TuzhanTrade> queryPage(TuzhanTrade tuzhanTrade, int pageNum, int pageSize) { PageHelper.startPage(pageNum, pageSize); List<TuzhanTrade> tuzhanTradeList = tuzhanTradeMapper.query(tuzhanTrade); PageInfo<TuzhanTrade> pageInfo = new PageInfo<>(tuzhanTradeList); return pageInfo; } </pre>	需要实现父类中的 <code>PageInfo<TuzhanTrade> queryPage(TuzhanTrade openshopTrade, int pageNum, int pageSize);</code> 方法，进行订单列表的查询操作
saveOrUpdateTrade	<pre> @Override public void saveOrUpdateTrade(JSONObject trade, String originContent, String shopCode) { } </pre>	需要实现父类中的 <code>void saveOrUpdateTrade(JSONObject trade, String originContent, String shopCode);</code> 方法，进行订单 trade 的保存操作
downloadPage	<pre> @Override public Result<Object> downloadPage(PlatformTradeOriginalMessageDTO platformTradeOriginalMessageDTO) { } </pre>	需要实现父类中的 <code>default Result<Object> downloadPage(T t) { return Result.error("此平台暂不支持批量下载"); }</code> 方法，进行订单列表的下载

download	<pre>@Override public boolean download(PlatformTradeOriginalMessageDTO platformTradeOriginalMessageDTO) { }</pre>	<p>需要实现父类中的 boolean download(T t); 方法，进行订单详情的下载</p>
formatKey	<pre>@Override public String formatKey() { return PlatformEnum.TUZHAN + "_" + FormatKeyEnum.FORMAT_TRADE.getCode(); }</pre>	<p>需要实现父类中 default String formatKey() { return FormatKeyEnum.class.getName(); } 方法，获取插入到 redis 中的 键值</p>
addRedis	<pre>@Override public void addRedis(String val) { platformOriginalFormatHandler.addRedis(formatKey(), val); }</pre>	<p>需要实现父类中的 default void addRedis(String val) { } 方法，保存到 redis 中</p>
getSourceJson	<pre>@Override public String getSourceJson(String tid) { }</pre>	<p>需要实现接口中的 String getSourceJson(String id); 方法，获取订单原始 json</p>
queryPage	<pre>@Override public PageInfo<PlatformTradeOriginalMessageDTO> queryPage(PlatformTradeOriginalMessageDTO platformTradeOriginalMessageDTO, int pageNum, int pageSize) { }</pre>	<p>需要实现接口中的 PageInfo<T> queryPage(T t, int pageNum, int pageSize); 方法，获取订单列表信息（分 页）</p>
format	<pre>@Override public Result<Object> format(List<String> tidList) { }</pre>	<p>需要实现接口中的 Result<Object> format(List<String> idList); 方法，进行订单的格式化操作</p>
queryPageFromPlatform	<pre>@Override public Result<Object> queryPageFromPlatform(PlatformTradeRequestDTO platformTradeRequestDTO) { }</pre>	<p>需要实现接口中的 default Result<Object> queryPageFromPlatform(T t) { return Result.error("平台暂 不支持列表查询"); } 方法，获取订单列表下载</p>
getFromPlatform	<pre>@Override</pre>	<p>需要实现接口中的</p>

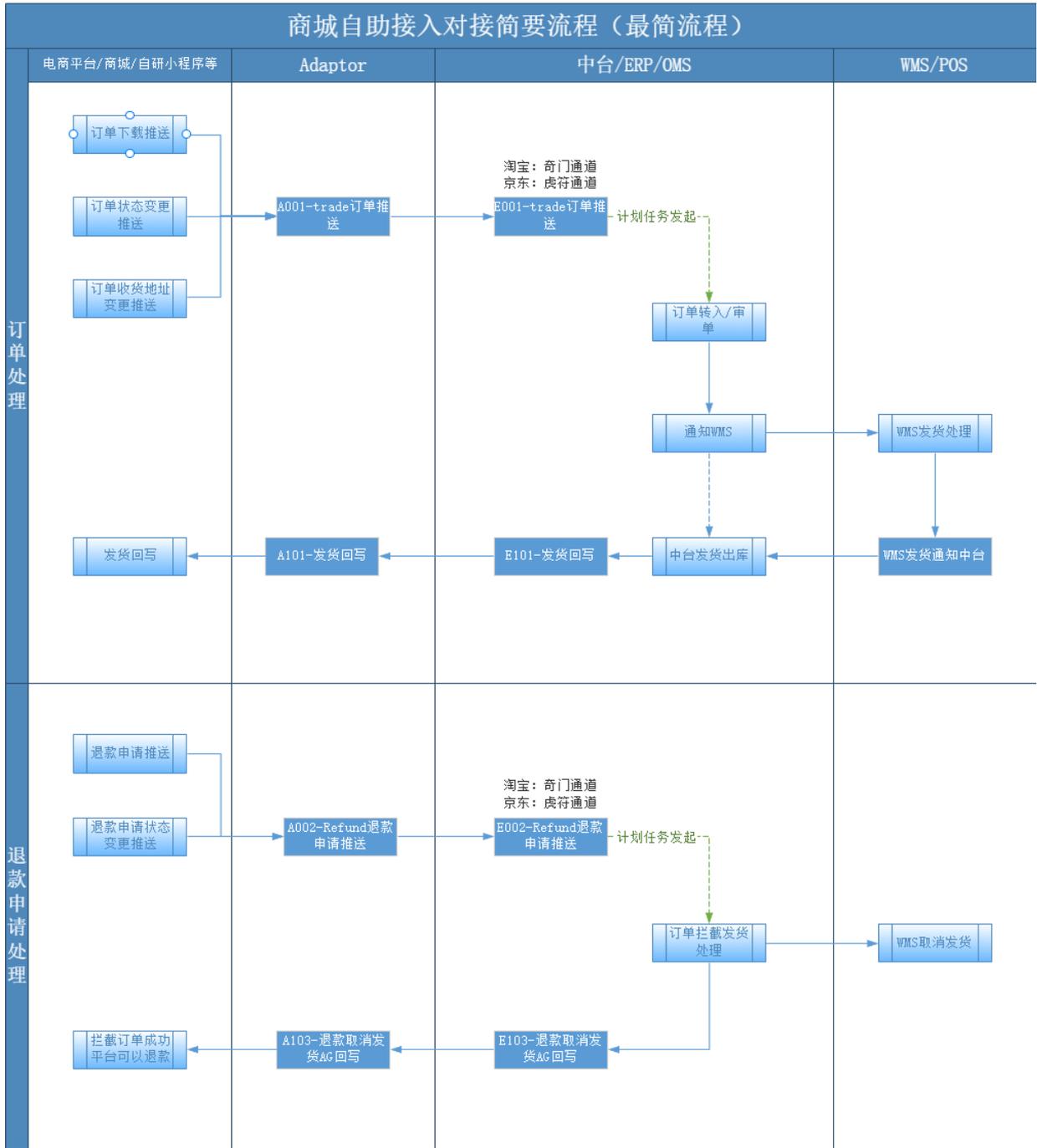
	<pre>public Result<Object> getFromPlatform(PlatformTradeRequestDTO platformTradeRequestDTO) { }</pre>	<pre>default Result<Object> getFromPlatform(T t) { return Result.error("平台暂 不支持查询详情"); } 方法，获取订单详情下载</pre>
shipWriteBack	<pre>@Override public Result<Object> shipWriteBack(Map<String, Object> paramMap) { }</pre>	<p>需要实现接口中的 Result<Object> shipWriteBack(Map<String, Object> map);</p> <p>方法，进行订单的发货回写操作</p>
split	<pre>@Override public Result<Object> split(Map<String, Object> map) { return Result.error("tuzhan 不支持京东拆分"); }</pre>	<p>需要实现接口中的 default Result<Object> split(Map<String, Object> map) { return Result.error("平台暂 不支持此方法"); }</p> <p>方法，进行订单的拆分回写平台（按需对接，目前只有京东支持拆分回写）</p>
updateToPlatform	<pre>@Override public Result<Object> updateToPlatform(Map<String, Object> map) { return Result.error("tuzhan 没有订单更新接口"); }</pre>	<p>需要实现接口中的 default Result<Object> updateToPlatform(Map<String, Object> map) { return Result.error("平台暂 不支持此订单信息更新"); }</p> <p>方法，进行订单的信息修改（按需对接）</p>

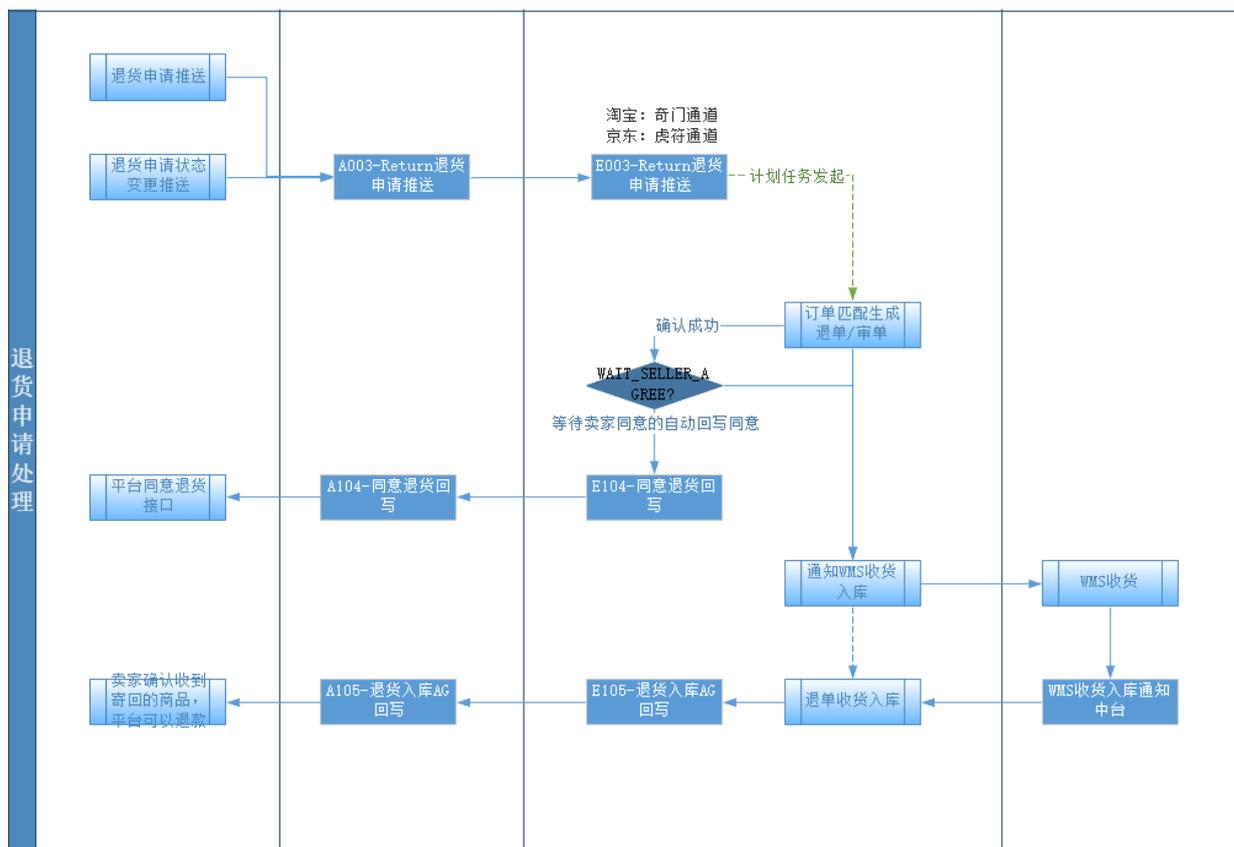
退款、退货、换货、商品等单据参考订单进行开发即可；

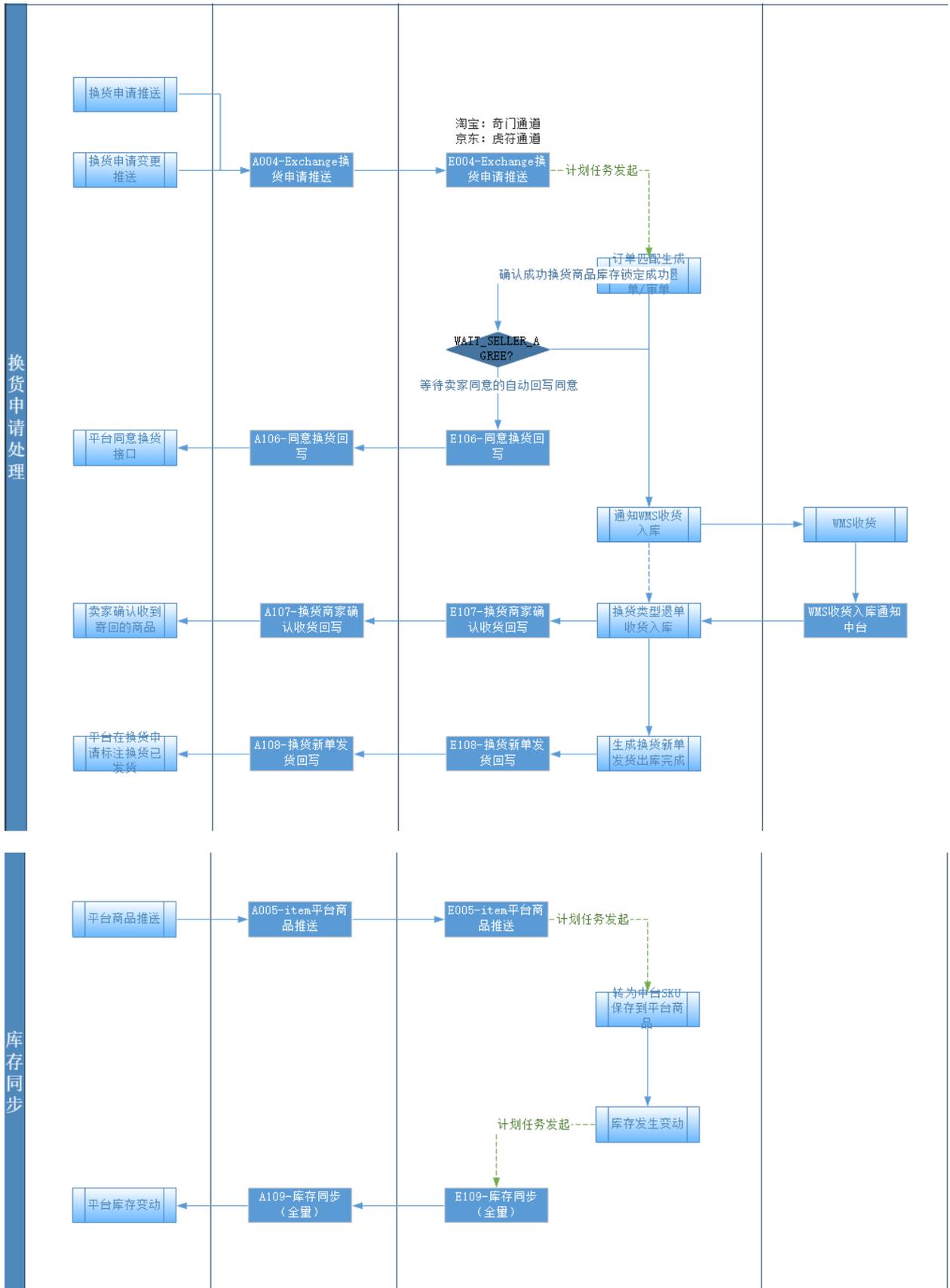
2.3.3. 商城自助接入 API

1. 商城自助接入对接简要流程（最简流程）

详情可查看在线流程说明：<https://www.processon.com/view/link/64905b29cdbc5438d2a7c312>

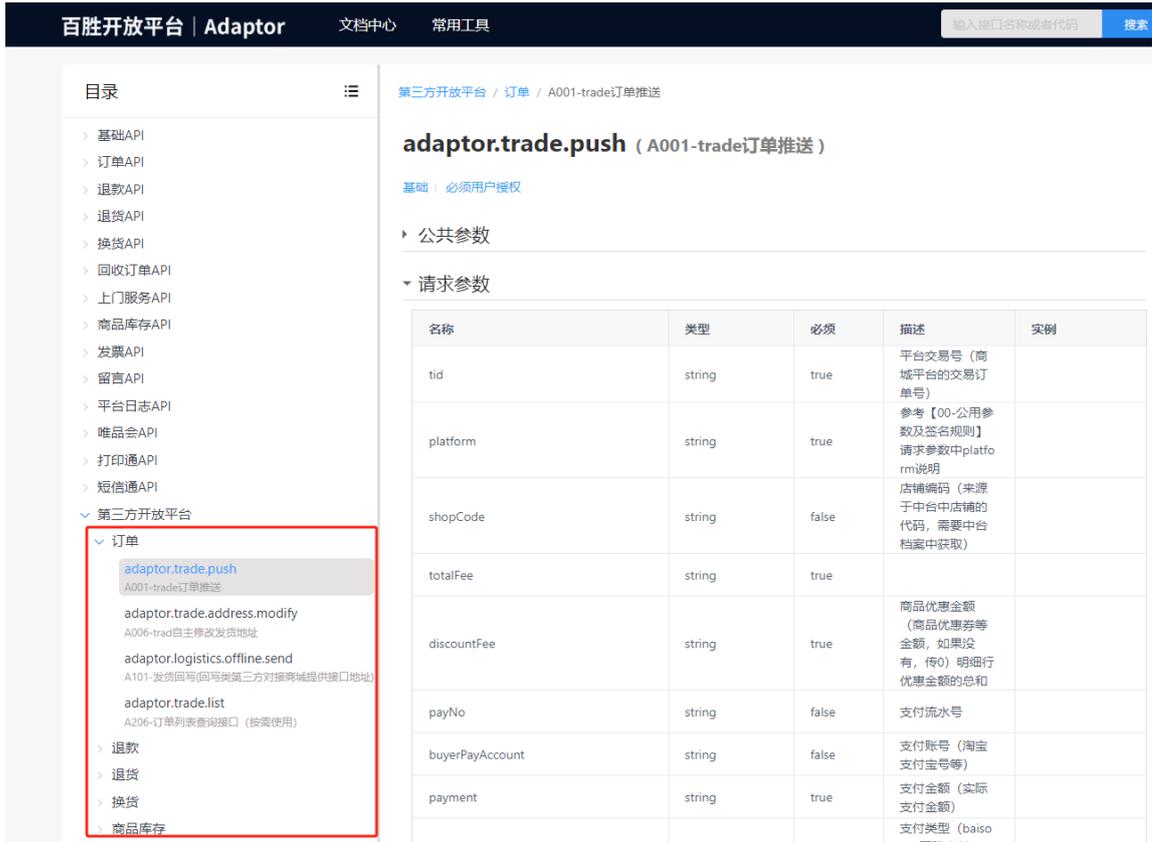






2. 接口说明

接口文档地址: <https://open.baison.net/>



3. 接入步骤

- a) 进入 adaptor public 管理界面新增第三方开放平台 appid, 配置第三方回写路径, 公钥私钥自动生成

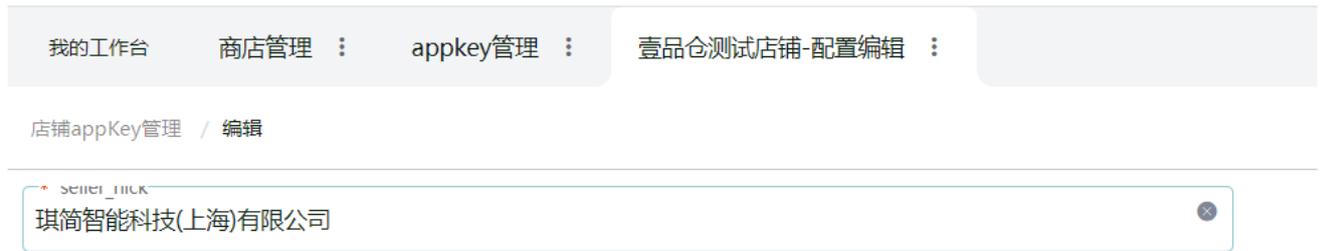




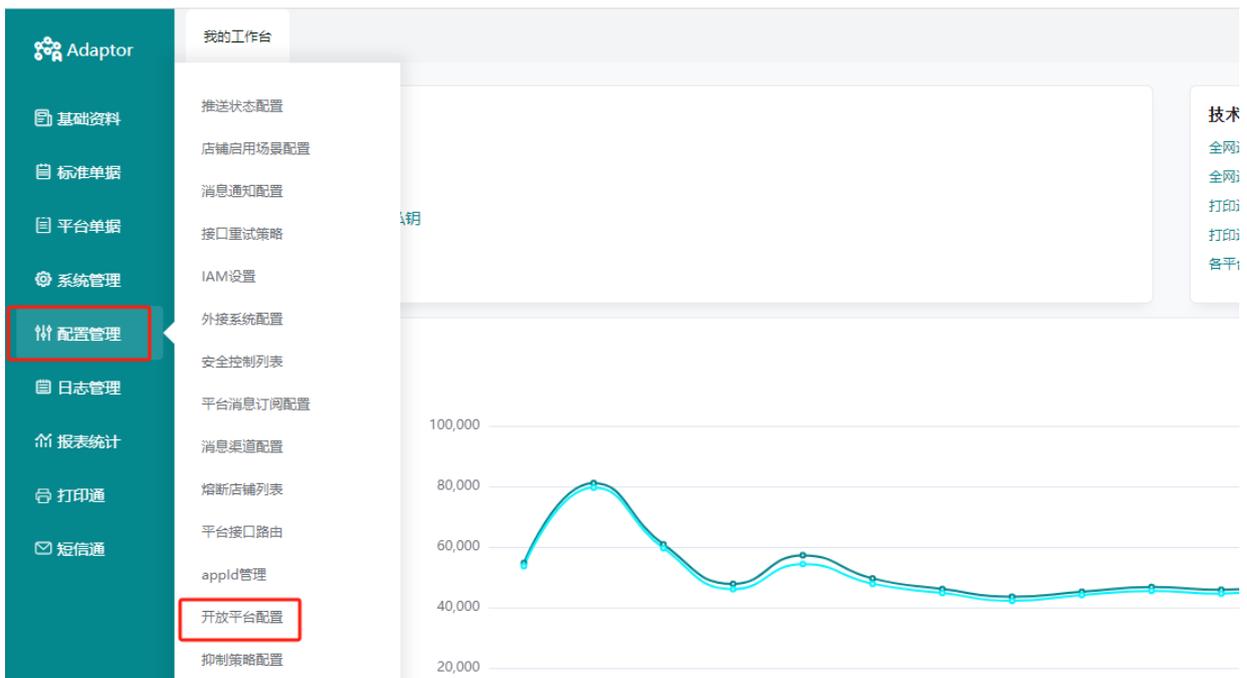
b) 使用当前租户登录 adaptor 系统，新建一个店铺，选择平台，填入店铺编号及店铺名称，点击保存



c) 到 appKey 管理中配置 seller_nick



d) 进入开放平台配置，商店授权，对刚才店铺授权





2.3.4. SDK 下载

详见附录

2.4. 平台接口调用优化

2.4.1. 抑制策略说明

场景一：Adaptor 自发调用平台接口及中台/ERP 在通过 Adaptor 系统请求平台接口的过程中，会有诸如 token 过期、服务不可用、参数错误及业务报错信息等；下游系统在调用的时候如果遇到错误都会有重试功能，会导致多次调用报错，平台接口调用统计失败率上升，继而影响平台 appkey 的评分；需要接口调用费用折扣是根据平台根据 appkey 的评分来进行设定的，如果评分降低，会增加公司的费用成本。

场景二：诸如库存同步场景，中台/ERP 中可以调度全量库存同步、调度增量库存同步、用户手动全量库存同步、用户手动增量库存同步等，还有一些单据进行锁定释放等的时候都会触发库存同步（库存实际未变动），此时有大量的相同的 sku 相同的同步数量多次请求平台进行库存同步，虽然都能调用成功，但是调用的量会成倍的增加，接口调用费用就会大幅上升，增加公司的费用成本。

基于以上场景，设计了抑制策略用以对请求进行拦截操作，并根据不同的策略进行不同的抑制处理。

2.4.2. 抑制策略场景

场景	描述
Token 过期	请求平台接口 token 过期之后，可以根据配置的关键字进行拦截，根据设置的抑制时间间隔进行抑制，如果在抑制时间间隔内，则不会进行平台接口调用
活动商品	主要针对淘宝促销活动的商品，在淘宝平台进行库存同步的时候，如果返回的报错信息中有“活动”字样，则会调用淘宝平台接口获取到当前商品涉及的活动信息（包括活动的开始时间、结束时间等）保存到系统中，下一次进行库存同步的时候会根据商品信息进行抑制，不再调用平台接口，直到活动结束后才会实际调用平台接口

库存同步	针对库存同步相同 SKU 相同数量多次同步的场景，可配置在 N 分钟（默认 1440 分钟即 24 小时）内只允许实际同步一次，大大降低无效库存同步	
其他	其他场景均根据平台返回的报错信息与系统中配置的关键字进行匹配，匹配成功则根据策略配置的时间间隔进行拦截	

抑制类型

抑制类型	说明	备注
时间间隔	在某一个时间间隔之内，进行抑制，一旦命中，在这个时间间隔之内，均不会再请求平台接口	开始时间：2023-10-12 09:00:00 结束时间：2023-10-12 10:00:00
时间段	不管是在哪一个时间点进行了抑制，结束时间均是从当前拦截的时间节点增加一个固定的时间段。比如现在 2023-10-12 13:00:00 触发了抑制，抑制时间段为 30 分钟，那么从当前时间开始一直到 2023-10-12 13:30:00 之间均不会再请求平台	时间段单位为分钟
Token 过期	对于 token 过期的场景，一旦触发 token 过期抑制，在随后的一段时间内（默认是 30 分钟），均不会再次请求平台接口，包括调度任务 api 拉取数据以及库存同步等被动场景接口	时间段单位为分钟

2.4.3. 抑制策略关键字

关键字在 public 中进行配置，配置完成之后，adaptor 中（SaaS 和套装）均会定时去 public 中获取配置的关键字到 adaptor 中进行抑制。

Adaptor 中可以单独配置抑制关键字，在 adaptor 中单独配置的关键字只针对于当前用户环境有效。

平台	编码	op 类型	时间间隔(分钟)	开始时间	结束时间	描述
douyinxiaodian	店铺未设置或启用售后小助手 erp 退货入仓自动同意退款策略	时间间隔	30			店铺未设置或启用售后小助手 erp 退货入仓自动同意退款策略
douyinxiaodian	没有能发货的订单	时间间隔	30			没有能发货的订单
douyinxiaodian	订单已完结，禁止发货	时间间隔	2048			订单已完结，禁止发货
douyinxiaodian	存在处理中流程，请勿同时重复请求	时间间隔	5			存在处理中流程，请勿同时重复请求

douyinxiaodian	售后状态与策略限制不符合, 未命中此策略	时间间隔	30			售后状态与策略限制不符合, 未命中此策略
douyinxiaodian	请检查售后单号是否正确	时间间隔	1440			请检查售后单号是否正确
douyinxiaodian	商品已删除, 不可更新	时间间隔	144000			商品已删除, 不可更新
douyinxiaodian	isv.parameter-invalid:7	时间段		2023/3/1 0:00	2031/4/1 0:00	系统错误: 商品已删除, 不可更新
douyinxiaodian	isv.parameter-invalid:5	时间段		2023/3/1 0:00	2032/4/1 0:00	找不到对应的 sku
douyinxiaodian	商品不存在	时间段		2023/4/24 0:00	2028/5/30 0:00	商品不存在
douyinxiaodian	正在参加平台活动 (抖音 618 好物节商品报名)	时间段		2023/6/1 0:00	2023/6/18 0:00	正在参加平台活动 (抖音 618 好物节商品报名)
douyinxiaodian	isv.access-token-expired	token 过期	30			access_token 已过期
douyinxiaodian	授权已被关闭, 请重新打开授权开关	token 过期	30			授权已被关闭, 请重新打开授权开关
douyinxiaodian	access_token 不存在	token 过期	30			access_token 不存在
fenixiao	推荐物流的订单状态为关闭状态, 无需发货处理	时间段		2022/12/1 0:00	2031/1/1 0:00	推荐物流的订单状态为关闭状态, 无需发货处理
fenixiao	产品 ID 不合法, 不能为空	时间段		2023/3/22 0:00	2030/4/30 0:00	产品 ID 不合法, 不能为空
fenixiao	非法或过期的 SessionKey 参数	token 过期	30			非法或过期的 SessionKey 参数
jingdong	400	时间段		2023/3/1 0:00	2031/4/1 0:00	SKU 不存在
jingdong	订单已出库	时间段	1	2023/3/27 0:00	2030/4/30 0:00	订单已出库
jingdong	token 已过期或者不存在	token 过期	30			token 已过期或者不存在
kuaihou	TOKEN 过期	token 过期	30			TOKEN 过期
pinduoduo	新疆、西藏专链商品不支持修改库存	时间间隔	1440000			新疆、西藏专链商品不支持修改库存
pinduoduo	订单已发货	时间间隔	14400			订单已发货
pinduoduo	不属于此商品, 无法更新	时间间隔	144000			不属于此商品, 无法更新
pinduoduo	订单已退款	时间间隔	14400			订单已退款
pinduoduo	access_token 已过期	token 过期	30			access_token 已过期
taobao	isv.ic_halo_rule_lite_item_edit_limit_1216_message	时间间隔	720			该商品已通过淘宝特价版活动审核。活动期间内, 禁止修改商品或 sku 信息, 如果需要退出活动请提交申请
taobao	OVERSEA_REFUNDED_INVALID	时间间隔	30			您的退货地址不规范, 买家售后易产生纠纷, 请填写规范的退货地址
taobao	该商品已被删除	时间间隔	7776000			该商品已被删除
taobao	isv.item-not-exist:invalid-numlid-or-iid	时间间隔	14400			商品 id 对应的商品不存在
taobao	isv.item-service-error:ITEM_NOT_FOUND-tmall	时间段		2022/12/1 0:00	2030/1/1 0:00	商品找不到

taobao	isv.item-quantity-update-sku-id-not-exist-tmall	时间段		2022/12/1 0:00	2030/1/1 0:00	商品 sku 不存在
taobao	isv.item-is-delete:invalid-numId-or-iid-tmall	时间段		2022/12/1 0:00	2030/1/1 0:00	商品 id 对应的商品已经被删除
taobao	isv.item-is-delete:invalid-numId-or-iid	时间段		2022/12/1 0:00	2030/1/1 0:00	商品 id 对应的商品已经被删除
taobao	isv.invalid-parameter-tmall	时间段		2023/1/1 0:00	2031/1/1 0:00	商品的 SKU 不存在
taobao	isv.item-not-exist:invalid-numId-or-iid	时间段		2023/1/1 0:00	2030/1/1 0:00	该商品不存在
taobao	isv.item-quantity-update-sku-id-not-exist	时间段		2023/1/1 0:00	2030/1/1 0:00	没有找到 SKUID
taobao	AT0112	时间段		2023/3/1 0:00	2031/4/1 0:00	推荐物流的订单状态为关闭状态, 无需发货处理
taobao	推荐物流的订单状态为关闭状态, 无需发货处理	时间段		2023/3/27 0:00	2030/4/30 0:00	推荐物流的订单状态为关闭状态, 无需发货处理
taobao	非法或过期的 SessionKey 参数	token 过期	30			非法或过期的 SessionKey 参数
taobao	获取商店授权失败, 请检查淘宝商店授权 Token 设置	token 过期	30			获取商店授权失败, 请检查淘宝商店授权 Token 设置
taobao	invalid-sessionkey	token 过期	30			非法或过期的 SessionKey 参数
taobao	用户没有对目标 APP KEY 进行授权	token 过期	30			用户没有对目标 APP KEY 进行授权
taobao	用户没有对目标 APP KEY 进行授权	token 过期	30			用户没有对目标 APP KEY 进行授权
taobao	用户没有对目标 APP KEY 进行授权	token 过期	30			用户没有对目标 APP KEY 进行授权
taobao	用户没有对目标 APP KEY 进行授权	token 过期	30			用户没有对目标 APP KEY 进行授权
taobao	用户没有对目标 APP KEY 进行授权	token 过期	30			用户没有对目标 APP KEY 进行授权
taobao	targetAppkey or customerId cannot be null at the same time	token 过期	30			targetAppkey or customerId cannot be null at the same time
taobao	targetAppkey or customerId cannot be null at the same time	token 过期	30			targetAppkey or customerId cannot be null at the same time
taobao	targetAppkey or customerId cannot be null at the same time	token 过期	30			targetAppkey or customerId cannot be null at the same time
taobao	targetAppkey or customerId cannot be null at the same time	token 过期	30			targetAppkey or customerId cannot be null at the same time
TP	5001	时间间隔	1440			订单已发货

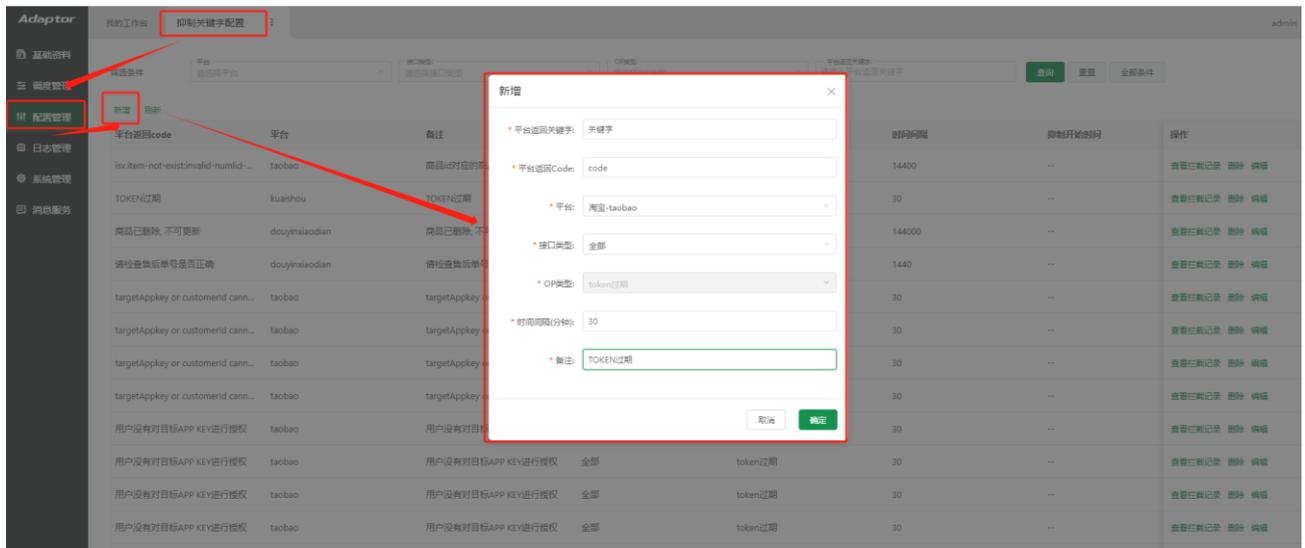
wxsphxd	当前商品状态不允许直接更新库存	时间间隔	10		当前商品状态不允许直接更新库存
---------	-----------------	------	----	--	-----------------

2.4.4. 抑制策略功能说明

1、public 中维护抑制的关键字

菜单：【配置管理】->【抑制关键字配置】

在 public 配置的抑制策略规则是全局生效的，adaptor 主动从 public 拉取的，默认每日凌晨 2 点执行，也可以在 adaptor 系统手动同步



2、public 查看拦截记录

在【抑制关键字配置】页面，点击【查看拦截记录】按钮进入到根据关键字查看拦截记录页面

抑制策略内容Code	店铺编码	店铺名称	租户Id	租户名称	seller_nick	授权编码	上报日期
相同SKU相同数量的不再同步	1600011902		suit			BCL24014455052015461158002	2023-10-11
相同SKU相同数量的不再同步	4996		suit				2023-10-11
相同SKU相同数量的不再同步	DVL8		suit			BCL68344847080873208565452	2023-10-11
相同SKU相同数量的不再同步	DNA2		suit			BCL68344847080873208565452	2023-10-11
相同SKU相同数量的不再同步	01158		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01442		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01242		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01241		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01240		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01188		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01096		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01083		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01070		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01019		85			BVLBSP821772a72d1008c0	2023-10-11
相同SKU相同数量的不再同步	01015		85			BVLBSP821772a72d1008c0	2023-10-11

租户id	租户名称	seller_nick	控制策略	上报日期	创建时间	拦截数量
suif			BCL24014455052015461158802	2023-10-11	2023-10-12 03:00:01	1697
suif				2023-10-11	2023-10-12 03:00:00	33
suif			BCL68344847000873208565432	2023-10-11	2023-10-12 03:00:01	11
suif			BCL68344847000873208565432	2023-10-11	2023-10-12 03:00:01	114
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	8565
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	2449
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	4376
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	658
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	1677
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	14101
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	5739
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	34106
85			BVL85P82172a72d1008c0	2023-10-11	2023-10-12 03:00:11	13438

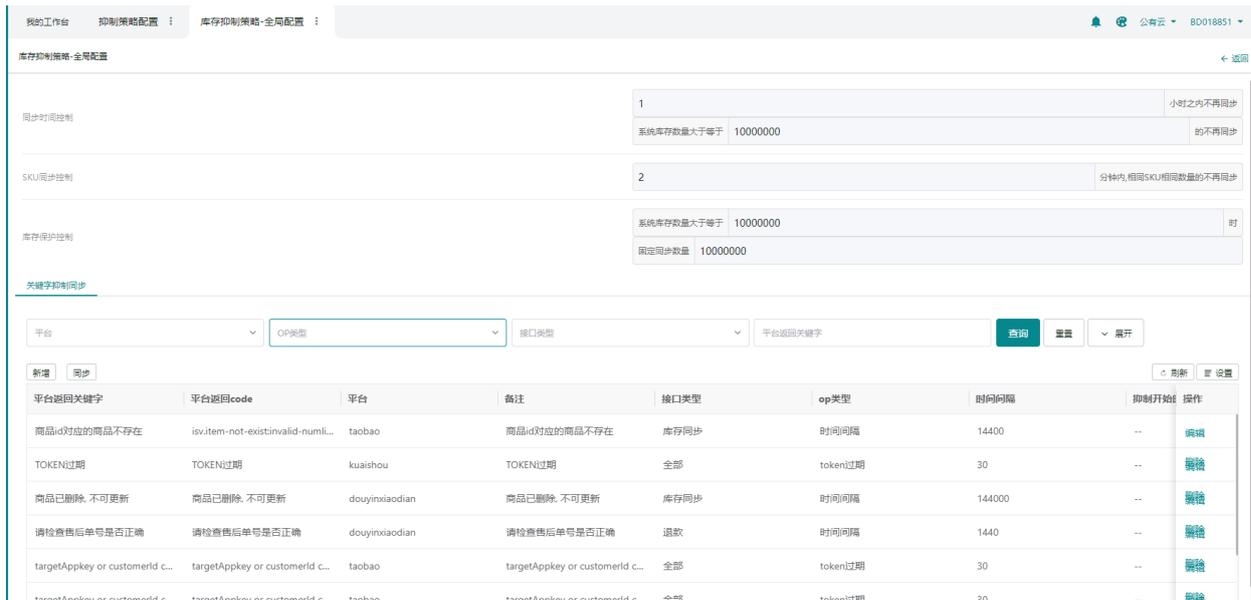
3、adaptor 中维护抑制关键字

菜单位置：【配置管理】->【抑制策略配置】

adaptor 中抑制关键字一般从 public 中会自动同步过来，

adaptor 中可以单独配置抑制关键字，在 adaptor 中单独配置的关键字只针对于当前用户环境有效。

oid	接口类型	抑制类型	活动名称	抑制开始时间	抑制结束时间	创建时间	创建人	操作
1344606941474	发货回号	其他类型的商品	订单已退款	2023-10-12 12:02:02	2023-10-22 12:02:02	2023-10-12 12:02:02		编辑
12479988871403425	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12482297579837342	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12473574838309966	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12473583525893895	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12467004184466624	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12484249742153370	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12484254435578935	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12493408364533522	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12464589790057562	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12479944968770668	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑
12489111681177484	发货回号	其他类型的商品	没有能发货的订单	2023-10-12 09:28:04	2023-10-12 09:58:04	2023-10-12 09:28:04		编辑



在 Adaptor 也可以手动触发从 public 中获取关键字，在【抑制策略-全局配置】页面的【关键字抑制同步】部分有【同步】按钮，点击按钮可以马上触发从 public 中同步

2.5. 编码规范手册

本手册以 Java 开发者为中心视角，划分为编程规约、异常日志、单元测试、安全规约、工程结构、MySQL 数据库六个维度，再根据内容特征，细分成若干二级子目录。

根据约束力强弱及故障敏感性，规约依次分为强制、推荐、参考三大类。对于规约条目的延伸信息中，“说明”对内容做了适当扩展和解释；“正例”提倡什么样的编码和实现方式；“反例”说明需要提防的雷区，以及真实的错误案例。

本手册的愿景是码出高效，码出质量。现代软件架构都需要协同开发完成，高效协作即降低协同成本，提升沟通效率，所谓无规矩不成方圆，无规范不能协作。

众所周知，制订交通法规表面上是要限制行车权，实际上是保障公众的人身安全。试想如果没有限速，没有红绿灯，谁还敢上路行驶。对软件来说，适当的规范和标准绝不是消灭代码内容的创造性、优雅性，而是限制过度个性化，以一种普遍认可的统一方式一起做事，提升协作效率。代码的字里行间流淌的是软件生命中的血液，质量的提升是尽可能少踩坑，杜绝踩重复的坑，切实提升质量意识。

本手册开发规范整体基于阿里巴巴 Java 开发手册，对个别不适用的规则进行了裁剪，并结合百胜的业务特点，适当进行了补充。

持续更新版本请关注：<http://bsconfluence.baison.net:40506/x/9bigAg>

2.5.1. 编程规约

2.5.1.1. 命名风格

1. 【强制】代码中的命名均不能以下划线或美元符号开始，也不能以下划线或美元符号结束。

反例：_name / __name / \$Object / name_ / name\$ / Object\$

2. 【强制】代码中的命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。

说明：正确的英文拼写和语法可以让读者易于理解，避免歧义。注意，即使纯拼音命名方式也要避免采用。

正例：alibaba / taobao / youku / hangzhou 等国际通用的名称，可视同英文。

反例：DaZhePromotion \[打折\] / getPingfenByName() \[评分\] / int 某变量 = 3

3. 【强制】类名使用 UpperCamelCase 风格，必须遵从驼峰形式，但以下情形例外：DO / BO / DTO / VO / AO

正例：MarcoPolo / UserDO / XmlService / TcpUdpDeal / TaPromotion

反例：macroPolo / UserDo / XMLService / TCPUDPDeal / TAPromotion

4. 【强制】方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风格，必须遵从驼峰形式。

正例：localValue / getHttpMessage() / inputUserId

5. 【强制】常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚，不要嫌名字长。

正例：MAX_STOCK_COUNT

反例：MAX_COUNT

6. 【强制】抽象类命名使用 Abstract 或 Base 开头；异常类命名使用 Exception 结尾；测试类命名以它要测试的类的名称开始，以 Test 结尾。

7. 【强制】中括号是数组类型的一部分，数组定义如下：String\[\] args;

反例：使用 String args\[\]的方式来定义。

8. 【强制】POJO 类中布尔类型的变量，都不要加 is，否则部分框架解析会引起序列化错误。

反例：定义为基本数据类型 Boolean isDeleted; 的属性，它的方法也是 isDeleted()，RPC 框架在反向解析的时候，“以为”对应的属性名称是 deleted，导致属性获取不到，进而抛出异常。

9. 【强制】包名统一使用小写，点分隔符之间有且仅有一个自然语义的英语单词。包名统一使用单数形式，但是类名如果有复数含义，类名可以使用复数形式。

正例：应用工具类包名为 com.alibaba.open.util、类名为 MessageUtils（此规则参考 spring 的框架结构）

10. 【强制】杜绝完全不规范的缩写，避免望文不知义。

反例：AbstractClass“缩写”命名成 AbsClass；condition“缩写”命名成 condi，此类随意缩写严重降低了代码的可阅读性。

11. 【推荐】为了达到代码自解释的目标，任何自定义编程元素在命名时，使用尽量完整的单词组合来表达其意。

正例：从远程仓库拉取代码的类命名为 PullCodeFromRemoteRepository。

反例：变量 `int a;` 的随意命名方式。

12. 【推荐】如果模块、接口、类、方法使用了设计模式，在命名时体现出具体模式。说明：将设计模式体现在名字中，有利于阅读者快速理解架构设计理念。

正例：`public class OrderFactory; public class LoginProxy; public class ResourceObserver;`

13. 【推荐】接口类中的方法和属性不要加任何修饰符号（`public` 也不要加），保持代码的简洁性，并加上有效的 Javadoc 注释。尽量不要在接口里定义变量，如果一定要定义变量，肯定是与接口方法相关，并且是整个应用的基础常量。

正例：接口方法签名：`void f();` 接口基础常量表示：`String COMPANY = "alibaba";`

反例：接口方法定义：`public abstract void f();` 说明：JDK8 中接口允许有默认实现，那么这个 `default` 方法，是对所有实现类都有价值的默认实现。

14. 接口和实现类的命名有两套规则：

1) 【强制】对于 `Service` 和 `DAO` 类，基于 SOA 的理念，暴露出来的服务一定是接口，内部的实现类用 `Impl` 的后缀与接口区别。

正例：`CacheServiceImpl` 实现 `CacheService` 接口。

2) 【推荐】如果是形容能力的接口名称，取对应的形容词做接口名（通常是 `-able` 的形式）。

正例：`AbstractTranslator` 实现 `Translatable`。 \\

15. 【参考】枚举类名建议带上 `Enum` 后缀，枚举成员名称需要全大写，单词间用下划线隔开。

说明：枚举其实就是特殊的常量类，且构造方法被默认强制是私有。

正例：枚举名字为 `ProcessStatusEnum` 的成员名称：`SUCCESS / UNKOWN_REASON`。

16. 【参考】各层命名规约：

A) `Service/DAO` 层方法命名规约

- 1) 获取单个对象的方法用 `get` 做前缀。
- 2) 获取多个对象的方法用 `list` 做前缀。
- 3) 获取统计值的方法用 `count` 做前缀。
- 4) 插入的方法用 `save/insert` 做前缀。
- 5) 删除的方法用 `remove/delete` 做前缀。
- 6) 修改的方法用 `update` 做前缀。

B) 领域模型命名规约

- 1) 数据对象：`xxxDO`，`xxx` 即为数据表名。
- 2) 数据传输对象：`xxxDTO`，`xxx` 为业务领域相关的名称。
- 3) 展示对象：`xxxVO`，`xxx` 一般为网页名称。
- 4) `POJO` 是 `DO/DTO/BO/VO` 的统称，禁止命名成 `xxxPOJO`。 \\

2.5.1.2. 常量定义

1. 【强制】不允许任何魔法值（即未经定义的常量）直接出现在代码中。

反例：

```
String key = "Id#taobao_" + tradeId;
cache.put(key, value);
```

2. 【强制】long 或者 Long 初始赋值时，使用大写的 L，不能是小写的 l，小写容易跟数字 1 混淆，造成误解。

说明：Long a = 2l；写的是数字的 2l，还是 Long 型的 2？

3. 【推荐】不要使用一个常量类维护所有常量，按常量功能进行归类，分开维护。

说明：大而全的常量类，非得使用查找功能才能定位到修改的常量，不利于理解和维护。

正例：缓存相关常量放在类 CacheConsts 下；系统配置相关常量放在类 ConfigConsts 下。

4. 【推荐】常量的复用层次有五层：跨应用共享常量、应用内共享常量、子工程内共享常量、包内共享常量、类内共享常量。

1) 跨应用共享常量：放置在二方库中，通常是 client.jar 中的 constant 目录下。

2) 应用内共享常量：放置在一方库中，通常是 modules 中的 constant 目录下。

反例：易懂变量也要统一定义成应用内共享常量，两位攻城师在两个类中分别定义了表示“是”的变量：

类 A 中：public static final String YES = "yes";

类 B 中：public static final String YES = "y"; A.YES.equals(B.YES)，预期是 true，但实际返回为 false，导致线上问题。

3) 子工程内部共享常量：即在当前子工程的 constant 目录下。

4) 包内共享常量：即在当前包下单独的 constant 目录下。

5) 类内共享常量：直接在类内部 private static final 定义。

5. 【推荐】如果变量值仅在一个范围内变化，且带有名称之外的延伸属性，定义为枚举类。下面正例中的数字就是延伸信息，表示星期几。

```
正例：public Enum { MONDAY(1), TUESDAY(2), WEDNESDAY(3), THURSDAY(4), FRIDAY(5), SATURDAY(6),
SUNDAY(7);}
```

2.5.1.3. 代码格式

1. 【强制】大括号的使用约定。如果是大括号内为空，则简洁地写成 {} 即可，不需要换行；如果是非空代码块则：

1) 左大括号前不换行。

2) 左大括号后换行。

3) 右大括号前换行。

4) 右大括号后还有 else 等代码则不换行；表示终止的右大括号后必须换行。

2. **【强制】** 左小括号和字符之间不出现空格；同样，右小括号和字符之间也不出现空格。详见 第 5 条下方正例提示。

反例：if (空格 a == b 空格)

3. **【强制】** if/for/while/switch/do 等保留字与括号之间都必须加空格。

4. **【强制】** 任何二目、三目运算符的左右两边都需要加一个空格。

说明：运算符包括赋值运算符=、逻辑运算符&&、加减乘除符号等。

5. **【强制】** 采用 4 个空格缩进，禁止使用 tab 字符。

说明：如果使用 tab 缩进，必须设置 1 个 tab 为 4 个空格。IDEA 设置 tab 为 4 个空格时，请勿勾选 Use tab character；而在 eclipse 中，必须勾选 insert spaces for tabs。

正例：（涉及 1-5 点）

```
public static void main(String[] args) {
    // 缩进 4 个空格
    String say = "hello";
    // 运算符的左右必须有一个空格
    int flag = 0;
    // 关键词 if 与括号之间必须有一个空格，括号内的 f 与左括号，0 与右括号不需要空格
    if (flag == 0) {
        System.out.println(say);
    }
    // 左大括号前加空格且不换行；左大括号后换行
    if (flag == 1) {
        System.out.println("world");
        // 右大括号前换行，右大括号后有 else，不用换行
    } else {
        System.out.println("ok");
        // 在右大括号后直接结束，则必须换行
    }
}
```

6. **【强制】** 注释的双斜线与注释内容之间有且仅有一个空格。

正例：// 注释内容，注意在//和注释内容之间有一个空格。

7. **【强制】** 单行字符数限制不超过 120 个，超出需要换行，换行时遵循如下原则：

- 1) 第二行相对第一行缩进 4 个空格，从第三行开始，不再继续缩进，参考示例。
- 2) 运算符与下文一起换行。

- 3) 方法调用的点符号与下文一起换行。
- 4) 方法调用时，多个参数，需要换行时，在逗号后进行。
- 5) 在括号前不要换行，见反例。

正例：

```
StringBuffer sb = new StringBuffer();  
// 超过 120 个字符的情况下，换行缩进 4 个空格，点号和方法名称一起换行  
sb.append("zi").append("xin")...  
.append("huang")...  
.append("huang")...  
.append("huang");
```

反例：

```
StringBuffer sb = new StringBuffer();  
// 超过 120 个字符的情况下，不要在括号前换行 sb.append("zi").append("xin")...append  
("huang");  
// 参数很多的方法调用可能超过 120 个字符，不要在逗号前换行  
method(args1, args2, args3, ...  
, argsX);
```

8. 【强制】方法参数在定义和传入时，多个参数逗号后边必须加空格。

正例：下例中实参的"a",后边必须要有一个空格。 `method("a", "b", "c");`

9. 【强制】IDE 的 text file encoding 设置为 UTF-8; IDE 中文件的换行符使用 Unix 格式，不要使用 Windows 格式。

10. 【推荐】没有必要增加若干空格来使某一行的字符与上一行对应位置的字符对齐。

正例： `int a = 3; long b = 4L; float c = 5F; StringBuffer sb = new StringBuffer();`

说明：增加 sb 这个变量，如果需要对齐，则给 a、b、c 都要增加几个空格，在变量比较多的情况下，是一种累赘的事情。

11. 【推荐】方法体内的执行语句组、变量的定义语句组、不同的业务逻辑之间或者不同的语义 之间插入一个空行。相同业务逻辑和语义之间不需要插入空行。

说明：没有必要插入多个空行进行隔开。

2.5.1.4. OOP 规约

1. 【强制】避免通过一个类的对象引用访问此类的静态变量或静态方法，无谓增加编译器解析成本，直接用类名来访问即可。
2. 【强制】所有的覆写方法，必须加@Override 注解。

说明: getObject()与 getObjcet()的问题。一个是字母的 0, 一个是数字的 0, 加@Override 可以准确判断是否覆盖成功。另外, 如果在抽象类中对方法签名进行修改, 其实现类会马上编译报错。

3. **【强制】**相同参数类型, 相同业务含义, 才可以使用 Java 的可变参数, 避免使用 Object。说明: 可变参数必须放置在参数列表的最后。(提倡同学们尽量不用可变参数编程)

正例: `public User getUsers(String type, Integer... ids) {...}`

4. **【强制】**外部正在调用或者二方库依赖的接口, 不允许修改方法签名, 避免对接口调用方产生影响。接口过时必须加@Deprecated 注解, 并清晰地说明采用的新接口或者新服务是什么。

5. **【强制】**不能使用过时的类或方法。

说明: java.net.URLDecoder 中的方法 decode(String encodeStr) 这个方法已经过时, 应该使用双参数 decode(String source, String encode)。接口提供方既然明确是过时接口, 那么有义务同时提供新的接口; 作为调用方来说, 有义务去考证过时方法的新实现是什么。

6. **【强制】**Object 的 equals 方法容易抛空指针异常, 应使用常量或确定有值的对象来调用 equals。

正例: `"test".equals(object)`; 反例: `object.equals("test")`;

说明: 推荐使用 java.util.Objects#equals (JDK7 引入的工具类)

7. **【强制】**所有的相同类型的包装类对象之间值的比较, 全部使用 equals 方法比较。

说明: 对于 Integer var = ? 在-128 至 127 范围内的赋值, Integer 对象是在 IntegerCache.cache 产生, 会复用已有对象, 这个区间内的 Integer 值可以直接使用==进行判断, 但是这个区间之外的所有数据, 都会在堆上产生, 并不会复用已有对象, 这是一个大坑, 推荐使用 equals 方法进行判断。

8. 关于基本数据类型与包装数据类型的使用标准如下:

- 1) **【强制】**所有的 POJO 类属性必须使用包装数据类型。
- 2) **【强制】**RPC 方法的返回值和参数必须使用包装数据类型。
- 3) **【推荐】**所有的局部变量使用基本数据类型。

说明: POJO 类属性没有初值是提醒使用者在需要使用时, 必须自己显式地进行赋值, 任何 NPE 问题, 或者入库检查, 都由使用者来保证。

正例: 数据库的查询结果可能是 null, 因为自动拆箱, 用基本数据类型接收有 NPE 风险。

反例: 比如显示成交总额涨跌情况, 即正负 x%, x 为基本数据类型, 调用的 RPC 服务, 调用不成功时, 返回的是默认值, 页面显示为 0%, 这是不合理的, 应该显示成中划线。所以包装数据类型的 null 值, 能够表示额外的信息, 如: 远程调用失败, 异常退出。

9. **【强制】**定义 DO/DTO/VO 等 POJO 类时, 不要设定任何属性默认值。

反例: POJO 类的 gmtCreate 默认值为 new Date();但是这个属性在数据提取时并没有置入具体值, 在更新其它字段时又附带更新了此字段, 导致创建时间被修改成当前时间。

10. **【强制】**序列化类新增属性时, 请不要修改 serialVersionUID 字段, 避免反序列化失败; 如果完全不兼容升级, 避免反序列化混乱, 那么请修改 serialVersionUID 值。

说明: 注意 serialVersionUID 不一致会抛出序列化运行时异常。

11. 【强制】构造方法里面禁止加入任何业务逻辑，如果有初始化逻辑，请放在 `init` 方法中。

12. 【强制】POJO 类必须写 `toString` 方法。使用 IDE 的中工具：`source> generate toString` 时，如果继承了另一个 POJO 类，注意在前面加一下 `super.toString`。

说明：在方法执行抛出异常时，可以直接调用 POJO 的 `toString()` 方法打印其属性值，便于排查问题。

13. 【推荐】使用索引访问用 `String` 的 `split` 方法得到的数组时，需做最后一个分隔符后有无内容的检查，否则会有抛 `IndexOutOfBoundsException` 的风险。

说明：

```
String str = "a,b,c,";
String[] ary = str.split(",");
// 预期大于 3，结果是 3
System.out.println(ary.length);
```

14. 【推荐】当一个类有多个构造方法，或者多个同名方法，这些方法应该按顺序放置在一起，便于阅读，此条规则优先于第 15 条规则。

15. 【推荐】类内方法定义顺序依次是：公有方法或保护方法 > 私有方法 > `getter/setter` 方法。

说明：公有方法是类的调用者和维护者最关心的方法，首屏展示最好；保护方法虽然只是子类关心，也可能是“模板设计模式”下的核心方法；而私有方法外部一般不需要特别关心，是一个黑盒实现；因为承载的信息价值较低，所有 `Service` 和 `DAO` 的 `getter/setter` 方法放在类体最后。

16. 【推荐】`setter` 方法中，参数名称与类成员变量名称一致，`this.成员名 = 参数名`。在 `getter/setter` 方法中，不要增加业务逻辑，增加排查问题的难度。

反例：

```
public Integer getData() {
    if (true) {
        return this.data + 100;
    } else {
        return this.data - 100;
    }
}
```

17. 【推荐】循环体内，字符串的连接方式，使用 `StringBuilder` 的 `append` 方法进行扩展。

说明：反编译出的字节码文件显示每次循环都会 `new` 出一个 `StringBuilder` 对象，然后进行 `append` 操作，最后通过 `toString` 方法返回 `String` 对象，造成内存资源浪费。

反例：

```
String str = "start";
for (int i = 0; i < 100; i++) {
    str = str + "hello";
}
```

```
}
```

18. 【推荐】final 可以声明类、成员变量、方法、以及本地变量，下列情况使用 final 关键字：

- 1) 不允许被继承的类，如：String 类。
- 2) 不允许修改引用的域对象，如：POJO 类的域变量。
- 3) 不允许被重写的方法，如：POJO 类的 setter 方法。
- 4) 不允许运行过程中重新赋值的局部变量。
- 5) 避免上下文重复使用一个变量，使用 final 描述可以强制重新定义一个变量，方便更好地进行重构。

19. 【推荐】慎用 Object 的 clone 方法来拷贝对象。

说明：对象的 clone 方法默认是浅拷贝，若想实现深拷贝需要重写 clone 方法实现属性对象的拷贝。

20. 【推荐】类成员与方法访问控制从严：

- 1) 如果不允许外部直接通过 new 来创建对象，那么构造方法必须是 private。
- 2) 工具类不允许有 public 或 default 构造方法。
- 3) 类非 static 成员变量并且与子类共享，必须是 protected。
- 4) 类非 static 成员变量并且仅在本类使用，必须是 private。
- 5) 类 static 成员变量如果仅在本类使用，必须是 private。
- 6) 若是 static 成员变量，必须考虑是否为 final。
- 7) 类成员方法只供类内部调用，必须是 private。
- 8) 类成员方法只对继承类公开，那么限制为 protected。

说明：任何类、方法、参数、变量，严控访问范围。过于宽泛的访问范围，不利于模块解耦。思考：如果是一个 private 的方法，想删除就删除，可是一个 public 的 service 方法，或者一个 public 的成员变量，删除一下，不得手心冒点汗吗？变量像自己的小孩，尽量在自己的视线内，变量作用域太大，无限制的到处跑，那么你会担心的。

2.5.1.5. 集合处理

1. 【强制】关于 hashCode 和 equals 的处理，遵循如下规则：

- 1) 只要重写 equals，就必须重写 hashCode。
- 2) 因为 Set 存储的是不重复的对象，依据 hashCode 和 equals 进行判断，所以 Set 存储的对象必须重写这两个方法。
- 3) 如果自定义对象做为 Map 的键，那么必须重写 hashCode 和 equals。

说明：String 重写了 hashCode 和 equals 方法，所以我们可以非常愉快地使用 String 对象作为 key 来使用。

2. 【强制】ArrayList 的 subList 结果不可强转成 ArrayList，否则会抛出 ClassCastException 异常，即 java.util.RandomAccessSubList cannot be cast to java.util.ArrayList.

说明：subList 返回的是 ArrayList 的内部类 SubList，并不是 ArrayList，而是 ArrayList 的一个视图，对于 SubList 子列表的所有操作最终会反映到原列表上。

3. **【强制】**在 subList 场景中，高度注意对原集合元素个数的修改，会导致子列表的遍历、增加、删除均会产生 ConcurrentModificationException 异常。

4. **【强制】**使用集合转数组的方法，必须使用集合的 toArray(T[] array)，传入的是类型完全一样的数组，大小就是 list.size()。

说明：使用 toArray 带参方法，入参分配的数组空间不够大时，toArray 方法内部将重新分配内存空间，并返回新数组地址；如果数组元素大于实际所需，下标为[list.size()]的数组元素将被置为 null，其它数组元素保持原值，因此最好将方法入参数组大小定义与集合元素个数一致。

正例：

```
List<String> list = new ArrayList<String>(2);
list.add("guan");
list.add("bao");
String[] array = new String[list.size()];
array = list.toArray(array);
```

反例：直接使用 toArray 无参方法存在问题，此方法返回值只能是 Object[]类，若强转其它类型数组将出现 ClassCastException 错误。

5. **【强制】**使用工具类 Arrays.asList()把数组转换成集合时，不能使用其修改集合相关的方法，它的 add/remove/clear 方法会抛出 UnsupportedOperationException 异常。

说明：asList 的返回对象是一个 Arrays 内部类，并没有实现集合的修改方法。Arrays.asList 体现的是适配器模式，只是转换接口，后台的数据仍是数组。

```
String[] str = new String[] { "you", "wu" };
List list = Arrays.asList(str);
```

第一种情况：list.add("yangguanbao"); 运行时异常。

第二种情况：str[0] = "gujin"; 那么 list.get(0)也会随之修改。

6. **【强制】**泛型通配符<? extends T>来接收返回的数据，此写法的泛型集合不能使用 add 方法，而<? super T>不能使用 get 方法，做为接口调用赋值时易出错。

说明：扩展说一下 PECS(Producer Extends Consumer Super)原则：第一、频繁往外读取内容的，适合用<? extends T>。第二、经常往里插入的，适合用<? super T>。

7. **【强制】**不要在 foreach 循环里进行元素的 remove/add 操作。remove 元素请使用 Iterator 方式，如果并发操作，需要对 Iterator 对象加锁。

正例：

```
Iterator<String> iterator = list.iterator();
while (iterator.hasNext()) {
```

```
String item = iterator.next();
if (删除元素的条件) {
    iterator.remove();
}
}
```

反例:

```
List<String> list = new ArrayList<String>();
list.add("1");
list.add("2");
for (String item : list) {
    if ("1".equals(item)) {
        list.remove(item);
    }
}
```

说明: 以上代码的执行结果肯定会出乎大家的意料, 那么试一下把"1"换成"2", 会是同样的结果吗?

8. **【强制】** 在 JDK7 版本及以上, Comparator 要满足如下三个条件, 不然 Arrays.sort, Collections.sort 会报 IllegalArgumentException 异常。

说明: 三个条件如下

- 1) x, y 的比较结果和 y, x 的比较结果相反。
- 2) $x > y, y > z$, 则 $x > z$ 。
- 3) $x = y$, 则 x, z 比较结果和 y, z 比较结果相同。

反例: 下例中没有处理相等的情况, 实际使用中可能会出现异常:

```
new Comparator<Student>() {
    @Override
    public int compare(Student o1, Student o2) {
        return o1.getId() > o2.getId() ? 1 : -1;
    }
};
```

9. **【推荐】** 集合初始化时, 指定集合初始值大小。

说明: HashMap 使用 HashMap(int initialCapacity) 初始化,

正例: $\text{initialCapacity} = (\text{需要存储的元素个数} / \text{负载因子}) + 1$ 。注意负载因子 (即 loader factor) 默认为 0.75, 如果暂时无法确定初始值大小, 请设置为 16 (即默认值)。

反例: HashMap 需要放置 1024 个元素, 由于没有设置容量初始大小, 随着元素不断增加, 容量 7 次被迫扩大, resize 需要重建 hash 表, 严重影响性能。

10. 【推荐】使用 entrySet 遍历 Map 类集合 KV，而不是 keySet 方式进行遍历。

说明：keySet 其实是遍历了 2 次，一次是转为 Iterator 对象，另一次是从 hashMap 中取出 key 所对应的 value。而 entrySet 只是遍历了一次就把 key 和 value 都放到了 entry 中，效率更高。如果是 JDK8，使用 Map.forEach 方法。

正例：values() 返回的是 V 值集合，是一个 list 集合对象；keySet() 返回的是 K 值集合，是一个 Set 集合对象；entrySet() 返回的是 K-V 值组合集合。

11. 【推荐】高度注意 Map 类集合 K/V 能不能存储 null 值的情况，如下表格：

集合类	Key	Value	Super	说明
Hashtable	不允许为 null	不允许为 null	Dictionary	线程安全
ConcurrentHashMap	不允许为 null	不允许为 null	AbstractMap	锁分段技术 (JDK8:CAS)
TreeMap	不允许为 null	允许为 null	AbstractMap	线程不安全
HashMap	允许为 null	允许为 null	AbstractMap	线程不安全

反例：由于 HashMap 的干扰，很多人认为 ConcurrentHashMap 是可以置入 null 值，而事实上，存储 null 值时会抛出 NPE 异常。

12. 【参考】合理利用好集合的有序性(sort)和稳定性(order)，避免集合的无序性(unsort)和 不稳定性(unorder)带来的负面影响。

说明：有序性是指遍历的结果是按某种比较规则依次排列的。稳定性指集合每次遍历的元素次序是一定的。如：ArrayList 是 order/unsort；HashMap 是 unorder/unsort；TreeSet 是 order/sort。

13. 【参考】利用 Set 元素唯一的特性，可以快速对一个集合进行去重操作，避免使用 List 的 contains 方法进行遍历、对比、去重操作。

2.5.1.6. 并发处理

1. 【强制】获取单例对象需要保证线程安全，其中的方法也要保证线程安全。

说明：资源驱动类、工具类、单例工厂类都需要注意。

2. 【强制】创建线程或线程池时请指定有意义的线程名称，方便出错时回溯。

正例：

```
public class TimerTaskThread extends Thread {
    public TimerTaskThread() {
        super.setName("TimerTaskThread");
        ...
    }
}
```

```
}  
}
```

3. 【强制】线程资源必须通过线程池提供，不允许在应用中自行显式创建线程。

说明：使用线程池的好处是减少在创建和销毁线程上所花的时间以及系统资源的开销，解决资源不足的问题。如果不使用线程池，有可能造成系统创建大量同类线程而导致消耗完内存或者“过度切换”的问题。

4. 【强制】线程池不允许使用 Executors 去创建，而是通过 ThreadPoolExecutor 的方式，这样的处理方式让写的同学更加明确线程池的运行规则，规避资源耗尽的风险。

说明：Executors 返回的线程池对象的弊端如下：

1) FixedThreadPool 和 SingleThreadPool：允许的请求队列长度为 Integer.MAX_VALUE，可能会堆积大量的请求，从而导致 OOM。

2) CachedThreadPool 和 ScheduledThreadPool：允许的创建线程数量为 Integer.MAX_VALUE，可能会创建大量的线程，从而导致 OOM。

5. 【强制】SimpleDateFormat 是线程不安全的类，一般不要定义为 static 变量，如果定义为 static，必须加锁，或者使用 DateUtils 工具类。

正例：注意线程安全，使用 DateUtils。亦推荐如下处理：

```
private static final ThreadLocal<DateFormat> df = new ThreadLocal<DateFormat>() {  
    @Override  
    protected DateFormat initialValue() {  
        return new SimpleDateFormat("yyyy-MM-dd");  
    }  
};
```

说明：如果是 JDK8 的应用，可以使用 Instant 代替 Date，LocalDateTime 代替 Calendar，

DateTimeFormatter 代替 SimpleDateFormat，官方给出的解释：simple beautiful strong immutable thread-safe。

13. 【参考】volatile 解决多线程内存不可见问题。对于一写多读，是可以解决变量同步问题，但是如果多写，同样无法解决线程安全问题。如果是 count++操作，使用如下类实现：AtomicInteger count = new AtomicInteger(); count.addAndGet(1); 如果是 JDK8，推荐使用 LongAdder 对象，比 AtomicLong 性能更好（减少乐观锁的重试次数）。

14. 【参考】HashMap 在容量不够进行 resize 时由于高并发可能出现死链，导致 CPU 飙升，在开发过程中可以使用其它数据结构或加锁来规避此风险。

15. 【参考】ThreadLocal 无法解决共享对象的更新问题，ThreadLocal 对象建议使用 static 修饰。这个变量是针对一个线程内所有操作共享的，所以设置为静态变量，所有此类实例共享此静态变量，也就是说在类第一次被使用时装载，只分配一块存储空间，所有此类的对象（只要是这个线程内定义的）都可以操控这个变量。

2.5.1.7. 控制语句

1. **【强制】** 在一个 switch 块内，每个 case 要么通过 break/return 等来终止，要么注释说明程序将继续执行到哪一个 case 为止；在一个 switch 块内，都必须包含一个 default 语句并且放在最后，即使它什么代码也没有。

2. **【强制】** 在 if/else/for/while/do 语句中必须使用大括号。即使只有一行代码，避免采用单行的编码方式：if (condition) statements;

3. **【推荐】** 表达异常的分支时，少用 if-else 方式，这种方式可以改写成：

```
if (condition) {  
    ...  
    return obj;  
}
```

// 接着写 else 的业务逻辑代码；

说明：如果非得使用 if()...else if()...else... 方式表达逻辑，**【强制】** 避免后续代码维护困难，请勿超过 3 层。

正例：超过 3 层的 if-else 的逻辑判断代码可以使用卫语句、策略模式、状态模式等来实现，其中卫语句示例如下：

```
public void today() {  
    if (isBusy()) {  
        System.out.println("change time.");  
        return;  
    }  
    if (isFree()) {  
        System.out.println("go to travel.");  
        return;  
    }  
    System.out.println("stay at home to learn Alibaba Java Coding Guidelines.");  
    return;  
}
```

4. **【推荐】** 除常用方法（如 getXxx/isXxx）等外，不要在条件判断中执行其它复杂的语句，将复杂逻辑判断的结果赋值给一个有意义的布尔变量名，以提高可读性。

说明：很多 if 语句内的逻辑相当复杂，阅读者需要分析条件表达式的最终结果，才能明确什么样的条件执行什么样的语句，那么，如果阅读者分析逻辑表达式错误呢？

正例：

// 伪代码如下

```
final boolean existed = (file.open(fileName, "w") != null) && (...) || (...);
if (existed) {
...
}
```

反例:

```
if ((file.open(fileName, "w") != null) && (...) || (...)) {
...
}
```

5. **【推荐】**循环体中的语句要考量性能，以下操作尽量移至循环体外处理，如定义对象、变量、获取数据库连接，进行不必要的 try-catch 操作（这个 try-catch 是否可以移至循环体外）。
6. **【推荐】**接口入参保护，这种场景常见的是用于做批量操作的接口。
7. **【参考】**下列情形，需要进行参数校验：
 - 1) 调用频次低的方法。
 - 2) 执行时间开销很大的方法。此情形中，参数校验时间几乎可以忽略不计，但如果因为参数错误导致中间执行回退，或者错误，那得不偿失。
 - 3) 需要极高稳定性和可用性的方法。
 - 4) 对外提供的开放接口，不管是 RPC/API/HTTP 接口。
 - 5) 敏感权限入口。
8. **【参考】**下列情形，不需要进行参数校验：
 - 1) 极有可能被循环调用的方法。但在方法说明里必须注明外部参数检查要求。
 - 2) 底层调用频度比较高的方法。毕竟是像纯净水过滤的最后一道，参数错误不太可能到底层才会暴露问题。一般 DAO 层与 Service 层都在同一个应用中，部署在同一台服务器中，所以 DAO 的参数校验，可以省略。
 - 3) 被声明成 private 只会被自己代码所调用的方法，如果能够确定调用方法的代码传入参数已经做过检查或者肯定不会有问题，此时可以不校验参数。

2.5.1.8. 注释规约

1. **【强制】**类、类属性、类方法的注释必须使用 Javadoc 规范，使用/**内容*/格式，不得使用 // xxx 方式。说明：在 IDE 编辑窗口中，Javadoc 方式会提示相关注释，生成 Javadoc 可以正确输出相应注释；在 IDE 中，工程调用方法时，不进入方法即可悬浮提示方法、参数、返回值的意义，提高阅读效率。
2. **【强制】**所有的抽象方法（包括接口中的方法）必须要用 Javadoc 注释、除了返回值、参数、异常说明外，还必须指出该方法做什么事情，实现什么功能。说明：对子类的实现要求，或者调用注意事项，请一并说明。

3. 【强制】所有的类都必须添加创建者和创建日期。
4. 【强制】方法内部单行注释，在被注释语句上方另起一行，使用//注释。方法内部多行注释使用/* */注释，注意与代码对齐。
5. 【强制】所有的枚举类型字段必须要有注释，说明每个数据项的用途。
6. 【推荐】与其“半吊子”英文来注释，不如用中文注释把问题说清楚。专有名词与关键字保持英文原文即可。
反例：“TCP 连接超时”解释成“传输控制协议连接超时”，理解反而费脑筋。
7. 【推荐】代码修改的同时，注释也要进行相应的修改，尤其是参数、返回值、异常、核心逻辑 等的修改。
说明：代码与注释更新不同步，就像路网与导航软件更新不同步一样，如果导航软件严重滞后，就失去了导航的意义。
8. 【参考】谨慎注释掉代码。在上方详细说明，而不是简单地注释掉。如果无用，则删除。
说明：代码被注释掉有两种可能性：
 - 1) 后续会恢复此段代码逻辑。
 - 2) 永久不用。前者如果没有备注信息，难以知晓注释动机。后者建议直接删掉（代码仓库保存了历史代码）。
9. 【参考】对于注释的要求：
 - 第一、能够准确反应设计思想和代码逻辑；
 - 第二、能够描述业务含义，使别的程序员能够迅速了解到代码背后的信息。完全没有注释的大段代码对于阅读者形同天书，注释是给自己看的，即使隔很长时间，也能清晰理解当时的思路；注释也是给继任者看的，使其能够快速接替自己的工作。
10. 【参考】好的命名、代码结构是自解释的，注释力求精简准确、表达到位。避免出现注释的一个极端：过多过滥的注释，代码的逻辑一旦修改，修改注释是相当大的负担。
反例：`// put elephant into fridge put(elephant, fridge);` 方法名 `put`，加上两个有意义的变量名 `elephant` 和 `fridge`，已经说明了这是在干什么，语义清晰的代码不需要额外的注释。
11. 【参考】特殊注释标记，请注明标记人与标记时间。注意及时处理这些标记，通过标记扫描，经常清理此类标记。线上故障有时候就是来源于这些标记处的代码。
 - 1) 待办事宜 (TODO)：(标记人, 标记时间, \[预计处理时间\]) 表示需要实现，但目前还未实现的功能。这实际上是一个 Javadoc 的标签，目前的 Javadoc 还没有实现，但已经被广泛使用。只能应用于类，接口和方法（因为它是一个 Javadoc 标签）。
 - 2) 错误，不能工作 (FIXME)：(标记人, 标记时间, \[预计处理时间\]) 在注释中用 `FIXME` 标记某代码是错误的，而且不能工作，需要及时纠正的情况。 \\

2.5.1.9. 其它规约

1. 【强制】在使用正则表达式时，利用好其预编译功能，可以有效加快正则匹配速度。

说明：不要在方法体内定义：`Pattern pattern = Pattern.compile(规则);`

2. **【强制】**velocity 调用 POJO 类的属性时，建议直接使用属性名取值即可，模板引擎会自动按规范调用 POJO 的 `getXxx()`，如果是 `boolean` 基本数据类型变量（`boolean` 命名不需要加 `is` 前缀），会自动调用 `isXxx()` 方法。

说明：注意如果是 `Boolean` 包装类对象，优先调用 `getXxx()` 的方法。

3. **【强制】**后台发送给页面的变量必须加 `!{var}`——中间的感叹号。

说明：如果 `var=null` 或者不存在，那么 `{var}` 会直接显示在页面上。

4. **【强制】**注意 `Math.random()` 这个方法返回是 `double` 类型，注意取值的范围 $0 \leq x < 1$ （能够取到零值，注意除零异常），如果想获取整数类型的随机数，不要将 `x` 放大 10 的若干倍然后取整，直接使用 `Random` 对象的 `nextInt` 或者 `nextLong` 方法。

5. **【强制】**获取当前毫秒数 `System.currentTimeMillis()`；而不是 `new Date().getTime()`；

说明：如果想获取更加精确的纳秒级时间值，使用 `System.nanoTime()` 的方式。在 JDK8 中，针对统计时间等场景，推荐使用 `Instant` 类。

6. **【推荐】**不要在视图模板中加入任何复杂的逻辑。

说明：根据 MVC 理论，视图的职责是展示，不要抢模型和控制器的活。

7. **【推荐】**任何数据结构的构造或初始化，都应指定大小，避免数据结构无限增长吃光内存。

8. **【推荐】**及时清理不再使用的代码段或配置信息。

说明：对于垃圾代码或过时配置，坚决清理干净，避免程序过度臃肿，代码冗余。

正例：对于暂时被注释掉，后续可能恢复使用的代码片断，在注释代码上方，统一规定使用三个斜杠 (`///`) 来说明注释掉代码的理由。

2.5.2. 异常日志

2.5.2.1. 异常处理

1. **【强制】**Java 类库中定义的一类 `RuntimeException` 可以通过预先检查进行规避，而不应该通过 `catch` 来处理，比如：`IndexOutOfBoundsException`，`NullPointerException` 等等。

说明：无法通过预检查的异常除外，如在解析一个外部传来的字符串形式数字时，通过 `catch` `NumberFormatException` 来实现。

正例：`if (obj != null) {...}`

反例：

```
try { obj.method() } catch (NullPointerException e) {...}
```

2. **【强制】**异常不要用来做流程控制，条件控制，因为异常的处理效率比条件分支低。

3. 【强制】对大段代码进行 try-catch，这是不负责任的表现。catch 时请分清稳定代码和非稳定代码，稳定代码指的是无论如何不会出错的代码。对于非稳定代码的 catch 尽可能进行区分异常类型，再做对应的异常处理。
4. 【强制】捕获异常是为了处理它，不要捕获了却什么都不处理而抛弃之，如果不想处理它，请将该异常抛给它的调用者。最外层的业务使用者，必须处理异常，将其转化为用户可以理解的内容。
5. 【强制】有 try 块放到了事务代码中，catch 异常后，如果需要回滚事务，一定要注意手动回滚事务。
6. 【强制】finally 块必须对资源对象、流对象进行关闭，有异常也要做 try-catch。

说明：如果 JDK7 及以上，可以使用 try-with-resources 方式。

7. 【强制】不能在 finally 块中使用 return，finally 块中的 return 返回后方法结束执行，不会再执行 try 块中的 return 语句。
8. 【强制】捕获异常与抛异常，必须是完全匹配，或者捕获异常是抛异常的父类。

说明：如果预期对方抛的是绣球，实际接到的是铅球，就会产生意外情况。

9. 【推荐】方法的返回值可以为 null，不强制返回空集合，或者空对象等，必须添加注释充分说明什么情况下会返回 null 值。调用方需要进行 null 判断防止 NPE 问题。

说明：本手册明确防止 NPE 是调用者的责任。即使被调用方法返回空集合或者空对象，对调用者来说，也并非高枕无忧，必须考虑到远程调用失败、序列化失败、运行时异常等场景返回 null 的情况。

10. 【推荐】防止 NPE，是程序员的基本修养，注意 NPE 产生的场景：

- 1) 返回类型为基本数据类型，return 包装数据类型的对象时，自动拆箱有可能产生 NPE。

反例：public int f() { return Integer 对象}，如果为 null，自动解箱抛 NPE。

- 2) 数据库的查询结果可能为 null。
- 3) 集合里的元素即使 isEmpty，取出的数据元素也可能为 null。
- 4) 远程调用返回对象时，一律要求进行空指针判断，防止 NPE。
- 5) 对于 Session 中获取的数据，建议 NPE 检查，避免空指针。
- 6) 级联调用 obj.getA().getB().getC()；一连串调用，易产生 NPE。

正例：使用 JDK8 的 Optional 类来防止 NPE 问题。

11. 【推荐】定义时区分 unchecked / checked 异常，避免直接抛出 new RuntimeException()，更不允许抛出 Exception 或者 Throwable，应使用有业务含义的自定义异常。推荐业界已定义过的自定义异常，如：DAOException / ServiceException 等。

12. 【参考】在代码中使用“抛异常”还是“返回错误码”，对于公司外的 http/api 开放接口必须使用“错误码”；而应用内部推荐异常抛出；跨应用间 RPC 调用优先考虑使用 Result 方式，封装 isSuccess() 方法、“错误码”、“错误简短信息”。

说明：关于 RPC 方法返回方式使用 Result 方式的理由：

- 1) 使用抛异常返回方式，调用方如果没有捕获到就会产生运行时错误。

2) 如果不加栈信息, 只是 new 自定义异常, 加入自己的理解的 error message, 对于调用端解决问题的帮助不会太多。如果加了栈信息, 在频繁调用出错的情况下, 数据序列化和传输的性能损耗也是问题。

13. 【参考】避免出现重复的代码 (Don't Repeat Yourself), 即 DRY 原则。

说明: 随意复制和粘贴代码, 必然会导致代码的重复, 在以后需要修改时, 需要修改所有的副本, 容易遗漏。必要时抽取共性方法, 或者抽象公共类, 甚至是组件化。

正例: 一个类中有多个 public 方法, 都需要进行数行相同的参数校验操作, 这个时候请抽取: private boolean checkParam(DTO dto) {...}

2.5.2.2. 日志规约

1. 【强制】应用中不可直接使用日志系统 (Log4j、Logback) 中的 API, 而应依赖使用日志框架 SLF4J 中的 API, 使用门面模式的日志框架, 有利于维护和各个类的日志处理方式统一。

```
import org.slf4j.Logger; import org.slf4j.LoggerFactory;

private static final Logger logger = LoggerFactory.getLogger(ABC.class);
```

2. 【强制】日志文件推荐至少保存 15 天, 因为有些异常具备以“周”为频次发生的特点。

3. 【强制】应用中的扩展日志 (如打点、临时监控、访问日志等) 命名方式: appName_logType_logName.log。

logType: 日志类型, 推荐分类有 stats/desc/monitor/visit 等;

logName: 日志描述。

这种命名的好处: 通过文件名就可知道日志文件属于什么应用, 什么类型, 什么目的, 也有利于归类查找。

正例: mppserver 应用中单独监控时区转换异常, 如: mppserver_monitor_timeZoneConvert.log

说明: 推荐对日志进行分类, 如将错误日志和业务日志分开存放, 便于开发人员查看, 也便于通过日志对系统进行及时监控。

4. 【强制】对 trace/debug/info 级别的日志输出, 必须使用条件输出形式或者使用占位符的方式。

说明: logger.debug("Processing trade with id: " + id + " and symbol: " + symbol); 如果日志级别是 warn, 上述日志不会打印, 但是会执行字符串拼接操作, 如果 symbol 是对象, 会执行 toString() 方法, 浪费了系统资源, 执行了上述操作, 最终日志却没有打印。

```
正例: (条件) if (logger.isDebugEnabled()) { logger.debug("Processing trade with id: " + id + "
and symbol: " + symbol); }
```

```
正例: (占位符) logger.debug("Processing trade with id: {} and symbol : {} ", id, symbol);
```

5. 【强制】避免重复打印日志, 浪费磁盘空间, 务必在 log4j.xml 中设置 additivity=false。

```
正例: <logger name="com.taobao.dubbo.config" additivity="false">
```

6. 【强制】异常信息应该包括两类信息: 案发现场信息和异常堆栈信息。如果不处理, 那么通过关键字 throws 往上抛出。

```
正例: logger.error(各类参数或者对象 toString + "_" + e.getMessage(), e);
```

7. 【推荐】谨慎地记录日志。生产环境禁止输出 debug 日志；有选择地输出 info 日志；如果使用 warn 来记录刚上线时的业务行为信息，一定要注意日志输出量的问题，避免把服务器磁盘撑爆，并记得及时删除这些观察日志。

说明：大量地输出无效日志，不利于系统性能提升，也不利于快速定位错误点。记录日志时请思考：这些日志真的有人看吗？看到这条日志你能做什么？能不能给问题排查带来好处？

8. 【参考】可以使用 warn 日志级别来记录用户输入参数错误的情况，避免用户投诉时，无所适从。

注意日志输出的级别，error 级别只记录系统逻辑出错、异常等重要的错误信息。如非必要，请不要在此场景打出 error 级别。

2.5.3. 单元测试

1. 【强制】好的单元测试必须遵守 AIR 原则。

说明：单元测试在线上运行时，感觉像空气（AIR）一样并不存在，但在测试质量的保障上，却是非常关键的。

好的单元测试宏观上来说，具有自动化、独立性、可重复执行的特点。

A: Automatic（自动化） I: Independent（独立性） R: Repeatable（可重复）

2. 【强制】单元测试应该是全自动执行的，并且非交互式的。测试框架通常是定期执行的，执行过程必须完全自动化才有意义。

输出结果需要人工检查的测试不是一个好的单元测试。

单元测试中不准使用 System.out 来进行人肉验证，必须使用 assert 来验证。

3. 【强制】保持单元测试的独立性。为了保证单元测试稳定可靠且便于维护，单元测试用例之间决不能互相调用，也不能依赖执行的先后次序。

反例：method2 需要依赖 method1 的执行，将执行结果做为 method2 的输入。

4. 【强制】单元测试是可以重复执行的，不能受到外界环境的影响。

说明：单元测试通常会被放到持续集成中，每次有代码 check in 时单元测试都会被执行。如果单测对外部环境（网络、服务、中间件等）有依赖，容易导致持续集成机制的不可用。

正例：为了不受外界环境影响，要求设计代码时就把 SUT 的依赖改成注入，在测试时用 spring 这样的 DI 框架注入一个本地（内存）实现或者 Mock 实现。

5. 【强制】对于单元测试，要保证测试粒度足够小，有助于精确定位问题。单测粒度至多是类级别，一般是方法级别。

说明：只有测试粒度小才能在出错时尽快定位到出错位置。单测不负责检查跨类或者跨系统的交互逻辑，那是集成测试的领域。

6. 【强制】核心业务、核心应用、核心模块的增量代码确保单元测试通过。

说明：新增代码及时补充单元测试，如果新增代码影响了原有单元测试，请及时修正。

7. 【强制】单元测试代码必须写在如下工程目录：src/test/java，不允许写在业务代码目录下。

说明：源码构建时会跳过此目录，而单元测试框架默认是扫描此目录。

8. 【推荐】单元测试的基本目标：语句覆盖率达到 70%；核心模块的语句覆盖率和分支覆盖率都要达到 100%

说明：在工程规约的应用分层中提到的 DAO 层，Manager 层，可重用度高的 Service，都应该进行单元测试。

9. 【推荐】编写单元测试代码遵守 BCDE 原则，以保证被测试模块的交付质量。

B: Border, 边界值测试，包括循环边界、特殊取值、特殊时间点、数据顺序等。

C: Correct, 正确的输入，并得到预期的结果。

D: Design, 与设计文档相结合，来编写单元测试。

E: Error, 强制错误信息输入（如：非法数据、异常流程、非业务允许输入等），并得到预期的结果。

10. 【推荐】对于数据库相关的查询，更新，删除等操作，不能假设数据库里的数据是存在的，或者直接操作数据库把数据插入进去，请使用程序插入或者导入数据的方式来准备数据。

反例：删除某一行数据的单元测试，在数据库中，先直接手动增加一行作为删除目标，但是这一行新增数据并不符合业务插入规则，导致测试结果异常。

11. 【推荐】和数据库相关的单元测试，可以设定自动回滚机制，不给数据库造成脏数据。或者对单元测试产生的数据有明确的前后缀标识。

正例：在 RDC 内部单元测试中，使用 RDC_UNIT_TEST_ 的前缀标识数据。

12. 【推荐】对于不可测的代码建议做必要的重构，使代码变得可测，避免为了达到测试要求而书写不规范测试代码。

13. 【推荐】在设计评审阶段，开发人员需要和测试人员一起确定单元测试范围，单元测试最好覆盖所有测试用例（UC）。

14. 【推荐】单元测试作为一种质量保障手段，不建议项目发布后补充单元测试用例，建议在项目提测前完成单元测试。

15. 【参考】为了方便地进行单元测试，业务代码应避免以下情况：

构造方法中做的事情过多。

存在过多的全局变量和静态方法。

存在过多的外部依赖。

存在过多的条件语句。

说明：多层条件语句建议使用卫语句、策略模式、状态模式等方式重构。

16. 【参考】不要对单元测试存在如下误解：

那是测试同学干的事情。本文是开发手册，凡是本文内容都是与开发同学强相关的。

单元测试代码是多余的。汽车的整体功能与各单元部件的测试正常与否是强相关的。

单元测试代码不需要维护。一年半载后，那么单元测试几乎处于废弃状态。

单元测试与线上故障没有辩证关系。好的单元测试能够最大限度地规避线上故障。

2.5.4. 安全规约

1. 【强制】隶属于用户个人的页面或者功能必须进行权限控制校验。说明：防止没有做水平权限校验就可随意访问、修改、删除别人的数据，比如查看他人的私信内容、修改他人的订单。

2. 【强制】用户敏感数据禁止直接展示，必须对展示数据进行脱敏。

说明：查看个人手机号码会显示成:158****9119，隐藏中间 4 位，防止隐私泄露。

3. 【强制】用户输入的 SQL 参数严格使用参数绑定或者 METADATA 字段值限定，防止 SQL 注入，禁止字符串拼接 SQL 访问数据库。

4. 【强制】用户请求传入的任何参数必须做有效性验证。

说明：忽略参数校验可能导致：

page size 过大导致内存溢出 恶意 order by 导致数据库慢查询 任意重定向 SQL 注入 反序列化注入 正则输入源串拒绝服务 ReDoS

说明：Java 代码用正则来验证客户端的输入，有些正则写法验证普通用户输入没有问题，但是如果攻击人员使用的是特殊构造的字符串来验证，有可能导致死循环的结果。

5. 【强制】禁止向 HTML 页面输出未经安全过滤或未正确转义的用户数据。

6. 【强制】表单、AJAX 提交必须执行 CSRF 安全过滤。

说明：CSRF(Cross-site request forgery)跨站请求伪造是一类常见编程漏洞。对于存在 CSRF 漏洞的应用/网站，攻击者可以事先构造好 URL，只要受害者用户一访问，后台便在用户不知情情况下对数据库中用户参数进行相应修改。

7. 【强制】在使用平台资源，譬如短信、邮件、电话、下单、支付，必须实现正确的防重放限制，如数量限制、疲劳度控制、验证码校验，避免被滥刷、资损。说明：如注册时发送验证码到手机，如果没有限制次数和频率，那么可以利用此功能骚扰到其它用户，并造成短信平台资源浪费。

8. 【推荐】发帖、评论、发送即时消息等用户生成内容的场景必须实现防刷、文本内容违禁词过滤等风控策略。

2.5.5. MySQL 数据库

2.5.5.1. 建表规约

1. 【强制】表达是与否概念的字段，必须使用 is_xxx 的方式命名，数据类型是 unsigned tinyint（1 表示是，0 表示否）。

说明：任何字段如果为非负数，必须是 unsigned。正例：表达逻辑删除的字段名 is_deleted，1 表示删除，0 表示未删除。

2. 【强制】表名、字段名必须使用小写字母或数字，禁止出现数字开头，禁止两个下划线中间只出现数字。数据库字段名的修改代价很大，因为无法进行预发布，所以字段名称需要慎重考虑。

说明：MySQL 在 Windows 下不区分大小写，但在 Linux 下默认是区分大小写。因此，数据库名、表名、字段名，都不允许出现任何大写字母，避免节外生枝。正例：aliyun_admin, rdc_config, level3_name 反例：

AliyunAdmin, rdcConfig, level_3_name

3. 【强制】表名不使用复数名词。

说明：表名应该仅仅表示表里面的实体内容，不应该表示实体数量，对应于 DO 类名也是单数形式，符合表达习惯。

4. 【强制】禁用保留字，如 desc、range、match、delayed 等，请参考 MySQL 官方保留字。

5. 【强制】主键索引名为 pk_字段名；唯一索引名为 uk_字段名；普通索引名则为 idx_字段名。

说明：pk_ 即 primary key；uk_ 即 unique key；idx_ 即 index 的简称。

6. 【强制】小数类型为 decimal，禁止使用 float 和 double。

说明：float 和 double 在存储的时候，存在精度损失的问题，很可能在值的比较时，得到不正确的结果。如果存储的数据范围超过 decimal 的范围，建议将数据拆成整数和小数分开存储。

7. 【强制】如果存储的字符串长度几乎相等，使用 char 定长字符串类型。

8. 【强制】varchar 是可变长字符串，不预先分配存储空间，长度不要超过 5000，如果存储长度大于此值，定义字段类型为 text，独立出来一张表，用主键来对应，避免影响其它字段索引效率。

9. 【强制】表必备 6 个字段：id, create_by, create_time, modify_by, modify_time, lastchanged。

说明：其中 id 必为主键，类型为 unsigned bigint、单表时自增、步长为 1。

create_time, modify_time 的类型均为 datetime 类型

create_by, modify_by 类型为 int，表示创建人、修改人的 id

lastchanged 类型为 timestamp，无需由程序处理。当数据发生变化时，由数据库自动更新，操作人，业务中统一存 code。

modify_time 创建时间同 create_time 一起赋值。

10. 【推荐】表的命名最好是加上“业务名称_表的作用”。

正例：alipay_task / force_project / trade_config

11. 【推荐】库名与应用名称尽量一致。

12. 【推荐】如果修改字段含义或对字段表示的状态追加时，需要及时更新字段注释。

13. 【推荐】字段允许适当冗余，以提高查询性能，但必须考虑数据一致。

冗余字段应遵循：1) 不是频繁修改的字段。2) 不是 varchar 超长字段，更不能是 text 字段。

正例：商品类目名称使用频率高，字段长度短，名称基本一成不变，可在相关联的表中冗余存储类目名称，避免关联查询。

14. 【推荐】单表行数超过 500 万行或者单表容量超过 2GB，才推荐进行分库分表。

说明：如果预计三年后的数据量根本达不到这个级别，请不要在创建表时就分库分表。

15. 【参考】合适的字符存储长度，不但节约数据库表空间、节约索引存储，更重要的是提升检索速度。

正例：如下表，其中无符号值可以避免误存负数，且扩大了表示范围。

人 150 岁之内 unsigned tinyint 1 无符号值：0 到 255

龟 数百岁 unsigned smallint 2 无符号值：0 到 65535

恐龙化石 数千万年 unsigned int 4 无符号值：0 到约 42.9 亿

太阳 约 50 亿年 unsigned bigint 8 无符号值：0 到约 10 的 19 次方

16, 建表时，数量、金额、价格类字段数据库给默认值为 0。单价和金额定义为：decimal(22, 6)。

2.5.5.2. 索引规约

1. 【强制】业务上具有唯一特性的字段，即使是多个字段的组合，也必须建成唯一索引。

说明：不要以为唯一索引影响了 insert 速度，这个速度损耗可以忽略，但提高查找速度是明显的；另外，即使在应用层做了非常完善的校验控制，只要没有唯一索引，根据墨菲定律，必然有脏数据产生。

2. 【强制】超过三个表禁止 join。需要 join 的字段，数据类型必须绝对一致；多表关联查询时，保证被关联的字段需要有索引。

说明：即使双表 join 也要注意表索引、SQL 性能。

3. 【强制】在 varchar 字段上建立索引时，必须指定索引长度，没必要对全字段建立索引，根据实际文本区分度决定索引长度即可。

说明：索引的长度与区分度是一对矛盾体，一般对字符串类型数据，长度为 20 的索引，区分度会高达 90%以上，可以使用 $\text{count}(\text{distinct left}(\text{列名}, \text{索引长度}))/\text{count}(\ast)$ 的区分度来确定。

4. 【强制】页面搜索严禁左模糊或者全模糊，如果需要请走搜索引擎来解决。

说明：索引文件具有 B-Tree 的最左前缀匹配特性，如果左边的值未确定，那么无法使用此索引。

5. 【推荐】如果有 order by 的场景，请注意利用索引的有序性。order by 最后的字段是组合索引的一部分，并且放在索引组合顺序的最后，避免出现 file_sort 的情况，影响查询性能。

正例：where a=? and b=? order by c; 索引：a_b_c

反例：索引中有范围查找，那么索引有序性无法利用，如：WHERE a>10 ORDER BY b; 索引 a_b 无法排序。

6. 【推荐】利用覆盖索引来进行查询操作，避免回表。

说明：如果一本书需要知道第 11 章是什么标题，会翻开第 11 章对应的那一页吗？目录浏览一下就好，这个目录就是起到覆盖索引的作用。

正例：能够建立索引的种类：主键索引、唯一索引、普通索引，而覆盖索引是一种查询的一种效果，用 explain 的结果，extra 列会出现：using index。

7. 【推荐】利用延迟关联或者子查询优化超多分页场景。

说明: MySQL 并不是跳过 offset 行, 而是取 offset+N 行, 然后返回放弃前 offset 行, 返回 N 行, 那当 offset 特别大的时候, 效率就非常的低下, 要么控制返回的总页数, 要么对超过 特定阈值的页数进行 SQL 改写。

正例: 先快速定位需要获取的 id 段, 然后再关联: `SELECT a.* FROM 表 1 a, (select id from 表 1 where 条件 LIMIT 100000,20) b where a.id=b.id`

8. 【推荐】 SQL 性能优化的目标: 至少要达到 range 级别, 要求是 ref 级别, 如果可以是 consts 最好。

说明: 1) consts 单表中最多只有一个匹配行 (主键或者唯一索引), 在优化阶段即可读取到数据。 2) ref 指的是使用普通的索引 (normal index)。 3) range 对索引进行范围检索。

反例: explain 表的结果, type=index, 索引物理文件全扫描, 速度非常慢, 这个 index 级别比较 range 还低, 与全表扫描是小巫见大巫。

9. 【推荐】建组合索引的时候, 区分度最高的在最左边。 正例: 如果 where a=? and b=? , a 列的几乎接近于唯一值, 那么只需要单建 idx_a 索引即可。

说明: 存在非等号和等号混合判断条件时, 在建索引时, 请把等号条件的列前置。如: where a>? and b=? 那么即使 a 的区分度更高, 也必须把 b 放在索引的最前列。

10. 【推荐】防止因字段类型不同造成的隐式转换, 导致索引失效。

11. 【参考】创建索引时避免有如下极端误解:

- 1) 宁滥勿缺。认为一个查询就需要建一个索引。
- 2) 宁缺勿滥。认为索引会消耗空间、严重拖慢更新和新增速度。
- 3) 抵制唯一索引。认为业务的唯一性一律需要在应用层通过“先查后插”方式解决。

(三) SQL 语句

1. 【强制】不要使用 count(列名)或 count(常量)来替代 count(*), count(*)是 SQL92 定义的标准统计行数的语法, 跟数据库无关, 跟 NULL 和非 NULL 无关。

说明: count(*)会统计值为 NULL 的行, 而 count(列名)不会统计此列为 NULL 值的行。

2. 【强制】count(distinct col) 计算该列除 NULL 之外的不重复行数, 注意 count(distinct col1, col2) 如果其中一列全为 NULL, 那么即使另一列有不同的值, 也返回为 0。

3. 【强制】当某一列的值全是 NULL 时, count(col)的返回结果为 0, 但 sum(col)的返回结果为 NULL, 因此使用 sum()时需注意 NPE 问题。

正例: 可以使用如下方式来避免 sum 的 NPE 问题: `SELECT IF(ISNULL(SUM(g)),0,SUM(g)) FROM table;`

4. 【强制】使用 ISNULL()来判断是否为 NULL 值。

说明: NULL 与任何值的直接比较都为 NULL。 1) NULL<>NULL 的返回结果是 NULL, 而不是 false。 2) NULL=NULL 的返回结果是 NULL, 而不是 true。 3) NULL<>1 的返回结果是 NULL, 而不是 true。

5. 【强制】在代码中写分页查询逻辑时, 若 count 为 0 应直接返回, 避免执行后面的分页语句。

6. 【强制】不得使用外键与级联, 一切外键概念必须在应用层解决。

说明：以学生和成绩的关系为例，学生表中的 student_id 是主键，那么成绩表中的 student_id 则为外键。如果更新学生表中的 student_id，同时触发成绩表中的 student_id 更新，即为级联更新。外键与级联更新适用于单机低并发，不适合分布式、高并发集群；级联更新是强阻塞，存在数据库更新风暴的风险；外键影响数据库的插入速度。

7. 【强制】禁止使用存储过程，存储过程难以调试和扩展，更没有移植性。

8. 【强制】数据订正时，删除和修改记录时，要先 select，避免出现误删除，确认无误才能执行更新语句。

9. 【推荐】in 操作能避免则避免，若实在避免不了，需要仔细评估 in 后边的集合元素数量，控制在 1000 个之内。

10. 【参考】如果有全球化需要，所有的字符存储与表示，均以 utf-8 编码，注意字符统计函数的区别。

说明：SELECT LENGTH("轻松工作")；返回为 12 SELECT CHARACTER_LENGTH("轻松工作")；返回为 4 如果需要存储表情，那么选择 utfmb4 来进行存储，注意它与 utf-8 编码的区别。

11. 【参考】TRUNCATE TABLE 比 DELETE 速度快，且使用的系统和事务日志资源少，但 TRUNCATE 无事务且不触发 trigger，有可能造成事故，故不建议在开发代码中使用此语句。

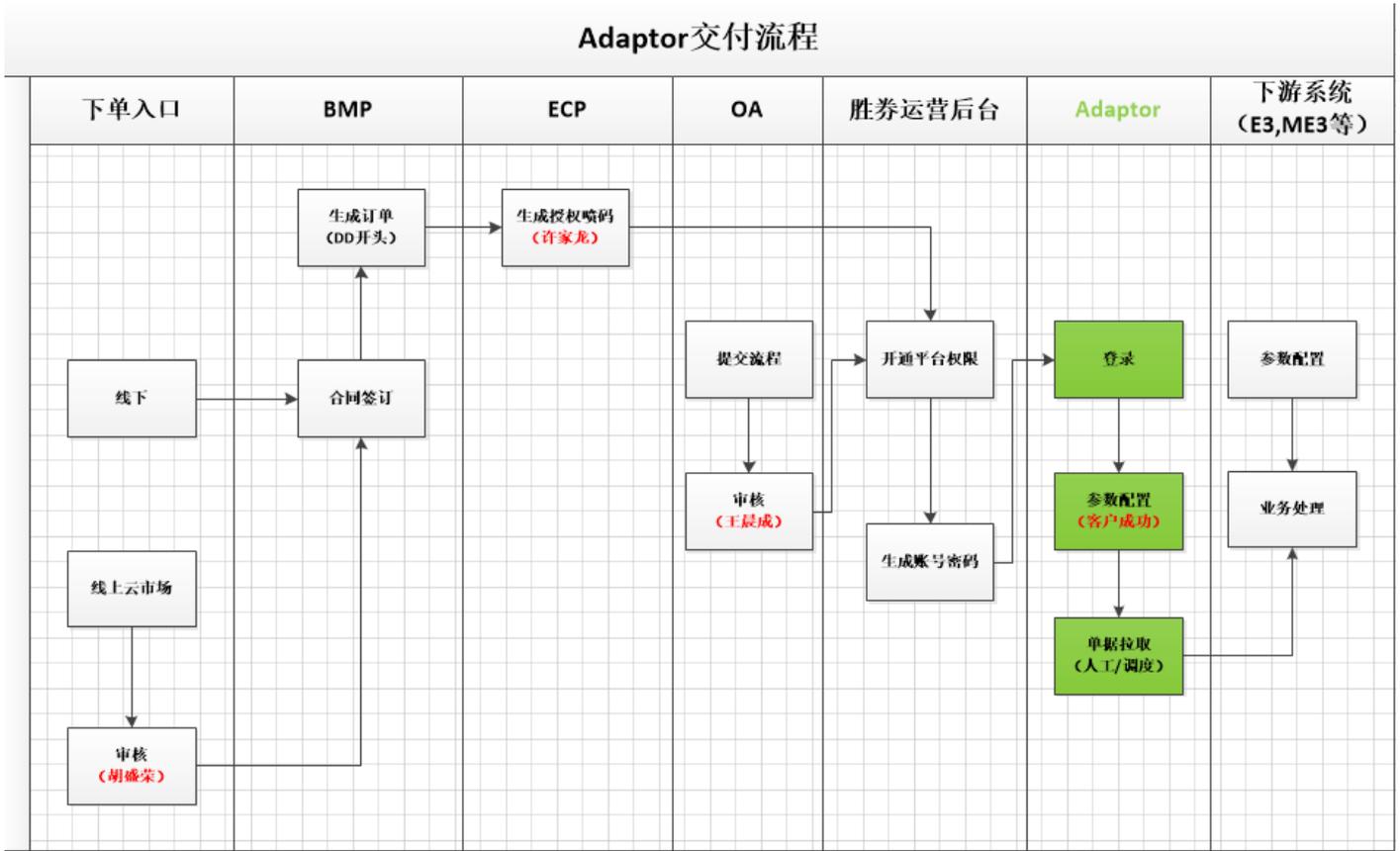
说明：TRUNCATE TABLE 在功能上与不带 WHERE 子句的 DELETE 语句相同。

三、运营篇

3.1. Adaptor 产品订购及授权开通流程

订购方式：

- 1、通过百胜云市场在线订购；
- 2、通过走线下合同订购对应产品；



3.1.1. 订购地址

订单通软件产品云市场订购地址: https://mall.baison.net/goods/rj?goods_sn=JSFW0030

打印通软件产品云市场订购地址: https://mall.baison.net/goods/rj?goods_sn=JSFW0029

短信通软件产品云市场订购地址: https://mall.baison.net/goods/rj?goods_sn=RJCP2023052600002

实施服务云市场订购地址: https://mall.baison.net/goods/fw?goods_sn=SSFW0001

订购过程有问题, 可联系运营人员-胡盛荣。

3.1.2. Adaptor 授权及账号开通

授权及账号开通:

1、云市场在线订购: 客户在云市场下单付款后, 运营审核后 BMP 自动生成 DD 开头的订单号, 找许家龙 ECP 授权, 然后提交 OA 申请开通账号即可;

2、线下合同订购: 合同签订客户付款后, 在 BMP 手动执行订单, 生成 DD 开头的订单号, 找对应的商务审核, 再找许家龙 ECP 授权, 然后提交 OA 申请开通账号即可;

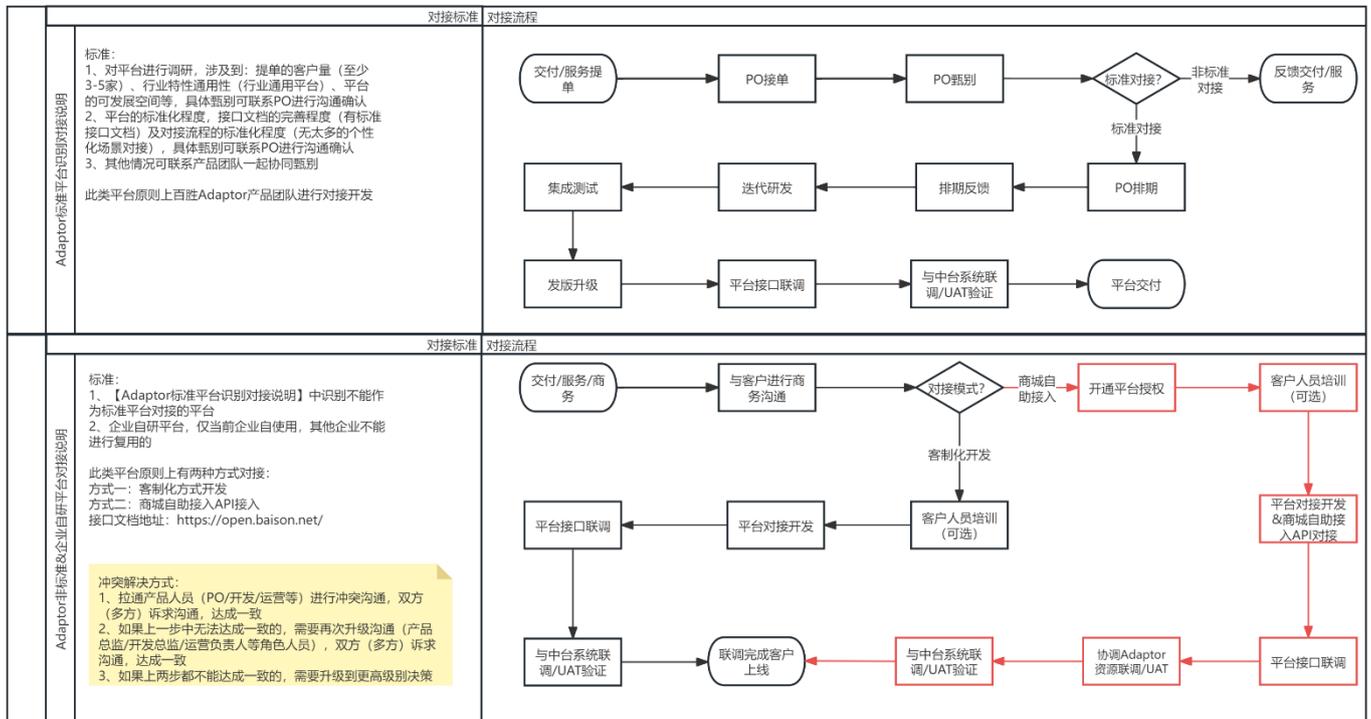
3.2. 计费模式

	KEY 归属	适用客户	资源归属	软件许可控制	费用组成	集群管理	持续发布	持续运维
年租共享资源	百胜 ISV KEY	客户愿意使用年租付费方式 百胜要求：客户单量一般	资源归属百胜，百胜进行购买	年租方式按平台授权	基础平台费用(E3+) 平台订阅费用	BSEIP	依托百胜发布 平台持续升级迭代	百胜运维
	百胜 ISV KEY	客户对性能要求比较高，对单据延迟比较敏感不能接受 百胜要求：客户单量较大	资源归属百胜，百胜进行购买	年租方式按平台授权	硬件费用（百胜采购） 基础平台费用(E3+) 平台订阅费用	BSEIP(简版)	依托百胜发布 平台持续升级迭代	百胜运维
年租独享资源	自研 KEY	客户有自研 KEY 客户对性能要求比较高，对单据延迟比较敏感不能接受	资源归属客户，百胜提供建议，客户进行购买	年租方式按平台授权	硬件费用（客户采购） 基础平台费用 平台订阅费用	BSEIP(简版)	依托百胜发布 平台持续升级迭代	百胜运维

3.3. 平台对接需求接入流程

在线文档地址：<https://www.processon.com/view/link/63ef493db6204d3d80122af3>

Adaptor新平台对接标准与流程



四、交付篇

4.1. Adaptor 配置

4.1.1. 标准接口平台

百胜开放平台地址：<https://open.baison.net/>

4.1.2. 配置说明

4.1.2.1. 系统配置

The screenshot shows the 'Adaptor' system configuration interface. On the left is a navigation menu with options: 基础资料, 标准单据, 平台单据, 系统管理 (selected), 配置管理, 日志管理, 报表统计, 打印通, 短信通. The main area is titled '我的工作台' and '参数配置'. It contains 15 numbered configuration items, each with a label, a value field, a description, and a '保存' (Save) button.

序号	配置项	值/状态	说明	操作
1	未登录时间超过	3	小时自动退出	保存
2	超过	60	天密码强制失效, 登录时提示"密码已过期, 请重新修改密码"	保存
3	授权喷码	BC...		保存
4	平台	wphjtx,dewu	订单格式化不进行物流进行映射	保存
5	日志是否存表	<input checked="" type="checkbox"/>		
6	订单发票是否推送开关	<input type="checkbox"/>		
7	淘宝换货格式化是否查商品明细	<input checked="" type="checkbox"/>		
8	淘宝订单金额是否扣减权益金	<input checked="" type="checkbox"/>		
9	推送状态开关	<input type="checkbox"/>		
10	店铺启用场景开关	<input type="checkbox"/>		
11	小红书merchantDiscount算入明细均摊金额中	<input type="checkbox"/>		
12	京东明细支付金额是否使用平台报文计算	<input type="checkbox"/>		
13	部分平台(美团等)订单是否保存运费	<input checked="" type="checkbox"/>		
14	京东是否拉取realpin字段	<input type="checkbox"/>		
15	数据推送方式(商品除外)	<input checked="" type="radio"/> 单个推送 <input type="radio"/> 批量推送		

- (1) 未登录时间超过 小时自动退出：登陆未操作的超时时间设置
- (2) 超过天密码强制失效，登录时提示“密码已过期，请重新修改密码”：密码强制修改设置
- (3) 授权喷码：百胜内部的客户唯一标识，用于平台 rds 查询、接口统计上报等使用

- (4) 外部访问地址：用于 bmp 等外部系统访问接口配置
- (5) 日志是否存表：回写平台接口日志开关，用于记录回调日志记录
- (6) 订单发票是否推送开关：配置是否进行发票推送
- (7) 淘宝订单金额是否扣减权益金：淘宝金额计算的时候，是否扣减权益金
- (8) 淘宝换货格式化是否查商品明细：淘宝换货查商品明细，找到对应的 outerskuid 值
- (9) 京东明细支付金额是否使用平台报文计算：已废弃，强制走平台报文计算
- (10) (淘宝)订单状态拉单和推送处理优先级：淘宝 rds 拉单，控制订单优先处理
- (11) 部分平台(美团等)订单是否保存运费：开启后，设置平台的运费为 0
- (12) 京东是否拉取 realpin 字段：京东订单是否需要 realpin 字段
- (13) 推送状态开关：与配置管理-推送状态配置配套使用，开启后会根据推送状态进行判断单据是否要推送
- (14) 店铺启用场景开关：与配置管理-店铺启用场景配置配套使用。控制拉单以及回写接口的调用
- (15) 数据推送方式(商品除外)：单据推送下游的方式，单个或者批量
- (16) 小红书 merchantDiscount 算入明细均摊金额中：小红书金额计算
- (17) 平台订单格式化不进行物流进行映射：哪些平台物流不映射成标准的物流

4.1.2.2. Nacos 配置

持续更新请关注：<http://bsconfluence.baison.net:40506/x/czXeC>

Group	Key (一级)	Key (二级)	说明
e3plus.adaptor	deleteHistoryBillData		是否删除历史数据
	debug		开启后，部分平台会模拟返回数据
	sign		前端调用接口是否开启验签
	pushToMq		是否需要推送消息到 mq，关闭后格式化后不会进入 mq
	backReturnSuccess		回写测试是否返回成功
	globalWriteBackLog		全局是否记录日志 可解除暂时不记录的判断(主要针对库存同步)
	overFormatTime		淘宝退款格式化失败超过一段时间后，设置成不格式化
	backDebug		开启后会回写拦截
	baisonRiskServerUrl		appkey 管理上，获取 code 默认跳转地址
adaptorUrl		saas 的地址，用于 bmp 回调 saas 的地址	

	mqRetryNum		推送单据重试次数
	pinduoduoConfig	arkEnable	是否开启拼多多方舟调用
		cloudAuthCode	是否调用参数要加入授权喷码，开启后走拼多多 rds 模式
	saas		是否为 saas 产品
	xxlRedisLock		拉单列表调用是否开启 redis 锁
	platListNotShow		该环境不需要展示的平台列表
	platListShow		该环境需要展示的平台列表
		doudianAppIdWhiteEnable	是否抖音小店做 ip 白名单上报
		cloudAuthCode	是否调用参数要加入授权喷码，开启后走抖音 rds 模式
		rdsByDdpModified	抖音 rds 订单查询的时候，时间采用 ddpModified 字段查询
	e3plus	url	私有化部署推送中台的地址
		appId	私有化部署 中台对应的 appid 值
		signType	私有化部署验签加密方式
		checkSign	下游请求是否开启验签
		appPrivateKey	私钥
		appPublicKey	公钥
	pub	url	公共服务地址，用于接口日志上报以及 saas 权限同步
		openIdUrl	第三方系统配置同步
	qimenPos	enable	淘宝订单是否走奇门
	login	onlyOneLogin	账号只能一处登陆
		maxErrorNumber	最大登陆次数
		modifyPwdTime	修改密码时间间隔，单位：天
		privateKey	私钥
		publicKey	公钥
	smsConfig	supportSms	是否支持短信功能
		operID	operID
		operPass	operPass
		url	发送短信地址

	signature	发送短信前缀
meituanConfig	notifyCheck	消息回调通知接口校验开关--美团平台配置地址时，会发送一个虚拟的请求
	poiReceived	是否开启收到订单时通知平台
	confirm	下单后直接走确认接口，防止超过 5 分钟限制
	format	是否开启推单直接格式化
	dazhongdianpingConfig	format
requestConfig	connectTimeout	代理接口连接超时时间
	readTimeout	代理接口读超时时间
	writeTimeout	代理接口写超时时间
messageCenter	routerAppKey	E3 电商 SPI 对接 APPKEY
	routerAppSecret	E3 电商 SPI 对接 APPSECRET
	appKey	E3 消息进来验签 appkey
	appSecret	E3 消息进来验签 appSecret
	e3OpenAppKey	E3 消息服务开通动作的验签 appkey
	e3OpenAppSecret	E3 消息服务开通动作的验签参数
	e3HttpUrl	调用 e3 消息的地址
	adaptorMnsAppKey	mns 服务验签 appkey
	adaptorMnsAppSecret	mns 服务验签 appSecret
	adaptorMnsHttpUrlPrefix	mns 调用地址
	platNotExistOldDataNotDelList	
defaultRetryDelayTime		标准单据重新推送总次数，下次重新推送时间间隔
resetRetryNumTime		重置推送重试次数，多久时间以前，单位为天
yace		是否进行压测
ipGetUrl		获取外网 ip 地址
searchMaxNum		调用 api 下载单据，列表下载默认最大个数
skiSaveTrade		跳过保存 trade 的表
skiPushE3		跳过保存推送中台

	batchPushNum		批量推送数量
	Ych	appKeyMap	appkey 对应的御城河 key
		ychKeyMap	御城河 key 对应的秘钥
iam	url		iam 请求地址
	app_id		adaptor 对应的 iam 的应用 id
	check		登陆是否需要 iam 验签
sq	url		胜券的请求地址
	appId		胜券的 appId
	appKey		胜券的 appkey

4.1.2.3. XXL-JOB 配置

持续更新请关注：<http://bsconfluence.baison.net:40506/x/cTXeC>

4.1.2.4. 百胜开放平台

中台/ERP 接入到 Adaptor，需要客户在百胜开放平台入驻，入驻资料经过审核之后，在百胜开放平台提交应用接入申请，申请提交之后运营人员审核通过之后，即可获得到 appId、公私钥信息等。

4.1.2.4.1. 首页

首页地址：<https://open.baison.net/>

接入流程：

接入流程



4.1.2.4.2. 账号

点击右上角登录按钮，跳转登录页面，百胜开放平台登录页面集成百胜云市场账号体系。输入云市场账号密码点击登录，会显示图像验证码，输入准确的图像验证码后，系统会发送短信验证码至认证手机，输入正确的验证码后即可登录。

百胜开放平台仅允许已完成企业认证的帐号申请应用。企业认证入口：云市场控制台-账号中心-实名认证，点击右侧“+注册企业”。请自行根据百胜开放平台流程进行认证。

4.1.2.4.3. 应用创建

在开发者账号注册并通过认证审核之后，登录到【我的工作台】，在我的工作台创建应用：

目前平台有两种类型的应用：

融合扩展应用：融合扩展是指开发者基于百胜软件产品扩展能力，对接产品内部开放 API，从而为企业开发定制应用。

集成对接应用：集成对接是指开发者对接百胜软件产品外部开放 API，从而实现异构系统之间的数据互通。需要申请的应用为【集成对接应用】。



×

创建应用

选择应用
 填写内容
 内容提交

选择应用

* 对接产品

* 授权喷码

* 回调URL

[查看示例](#)

填写内容

* 应用名称

* 联系人

* 手机号

* 邮箱

* 应用介绍

* 上传资料 (必填: 仅支持 pptx、pdf、doc、docx格式, 不超过 30 MB)

+

- 对接产品: 选择三方应用需要对接的百胜软件产品, 当前平台开放对接的产品有: E3OMS、E3ERP、Adaptor、ME3、E3+。**对接产品选择 Adaptor。**
- 授权喷码: 选择产品后, 系统会自动带出该认证企业该产品的授权喷码。如果客户是第一次使用 Adaptor, 需要发起钉钉 OA 流程。服务中心开通 Adaptor 环境后, 才能申请应用, 否则开放平台信息无法正常同步至 Adaptor 环境。
- 回调地址: 指需要对接的百胜产品的服务地址, 用于百胜开放平台向客户系统推送对接所需信息。选择产品后, 点击“示例”字样, 按照示例补充域名地址后填写。Adaptor SaaS 的回调地址为: <https://public-adaptor.baison.net/adaptonbsopen/notify>
- 应用名称: 填写需要对接百胜产品的三方应用名称, 建议应用名称能够体现出该应用的开发方和主要作用。例如: A 企业的自研 OMS 系统需要对接, 建议起名为: A 企业简称自研 OMS。
- 联系人: 填写客户方三方应用管理员的名称;
- 手机号: 填写客户方三方应用管理员的手机号;
- 邮箱: 填写三方应用管理员的邮箱, 应用审核通过后会发送审核通过邮件至该邮箱;

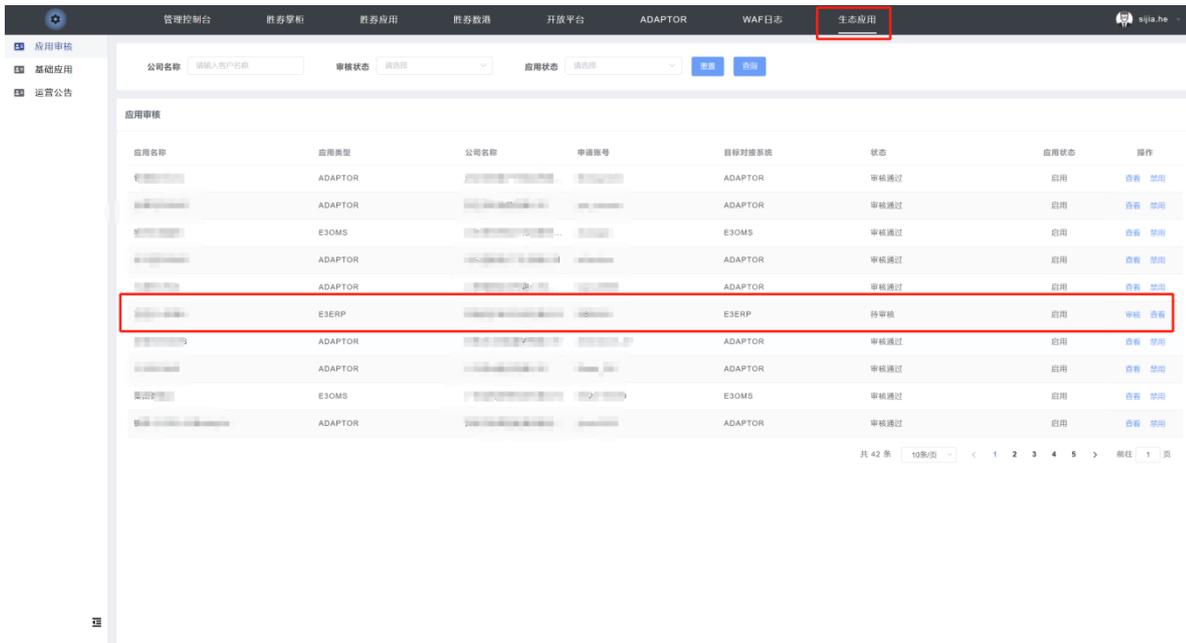
- 应用介绍：请简单描述三方应用对接百胜软件产品所需实现的场景；
- 上传资料：请上传三方应用的产品介绍资料。

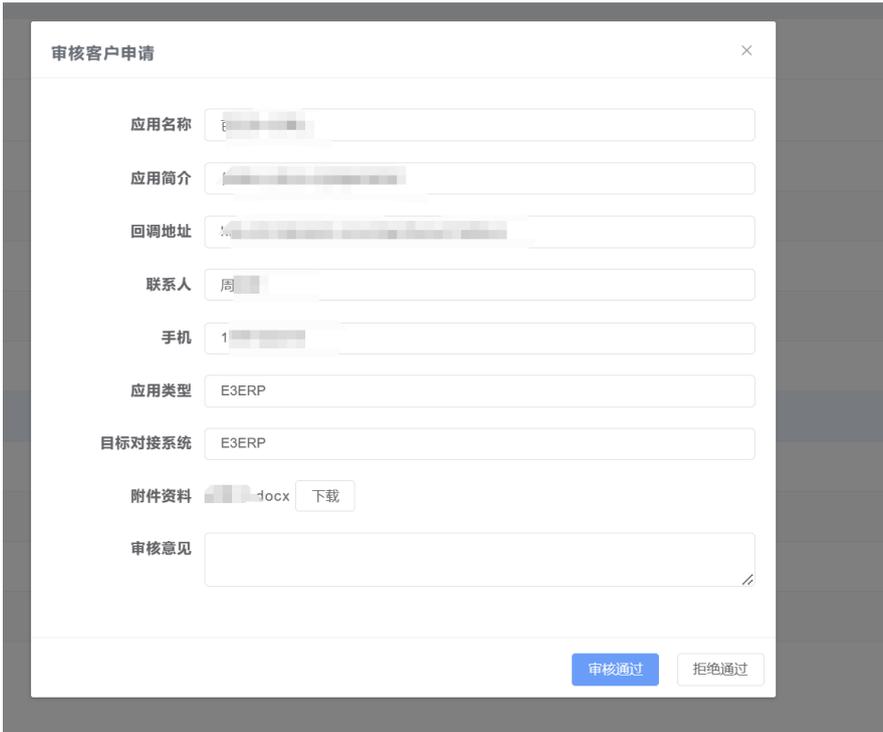
以上内容全部填写完毕后，点击确认提交，等待平台运营审核。

开放平台运营人员会在申请当日完成应用审核，如果比较急，可钉钉联系催办。审核通过后，平台会自动向之前应用创建时留下的邮箱发送审核通过的通知邮件。

4.1.2.4.4. 应用审核

开发者通过以上步骤提交应用申请后，开放平台运营人员可在胜券运营后台-生态应用-应用审核页面，对应用进行审核，或者驳回操作。如驳回申请，需要填写告知申请人驳回理由。

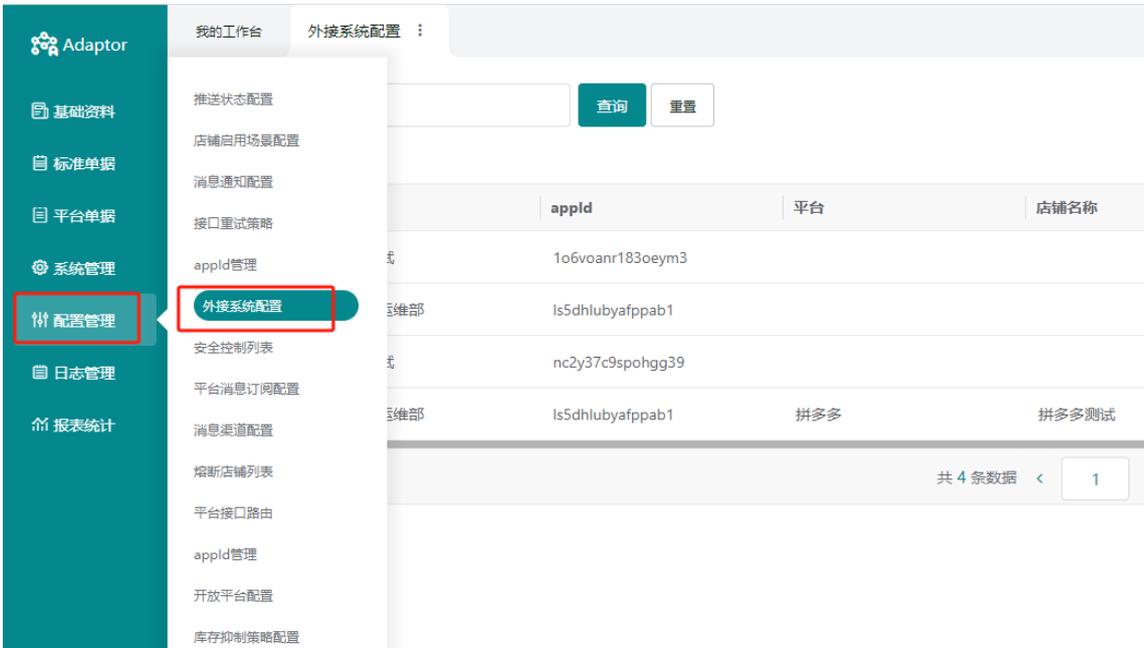




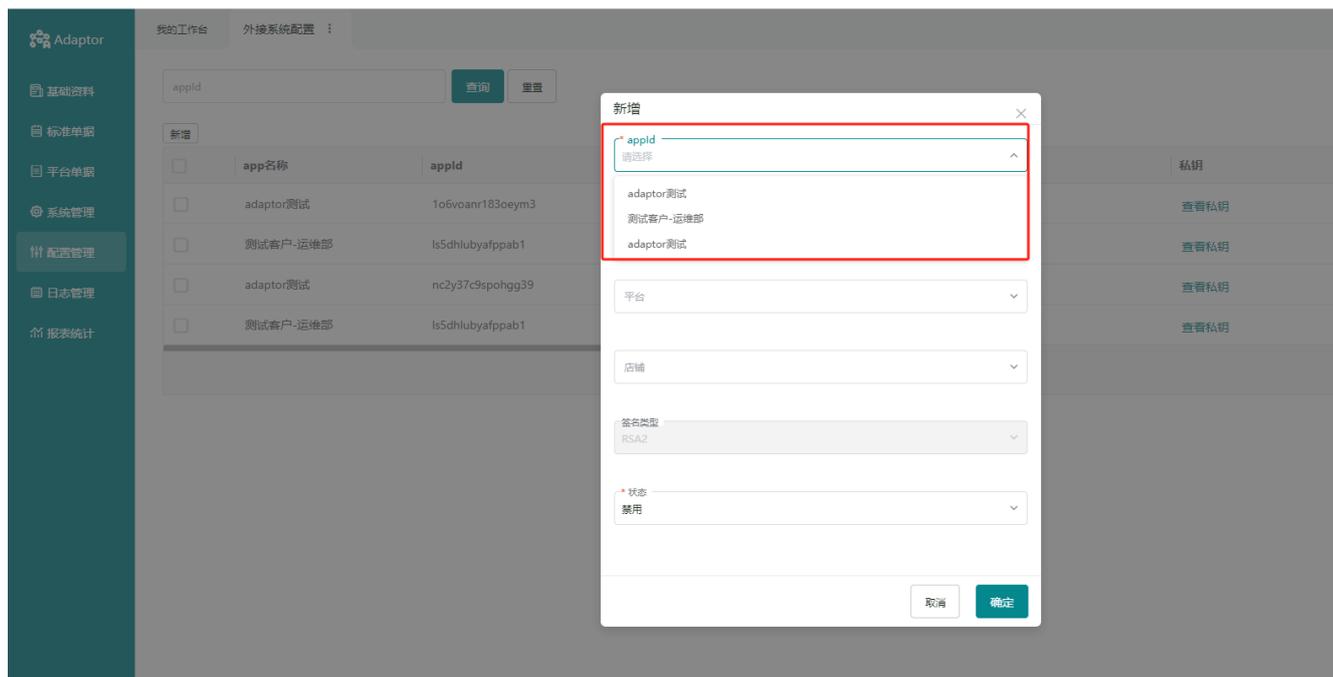
应用审核通过之后，即可在百胜开放平台【我的工作台】中查看对应的 appId、公私钥信息。

4.1.2.5. Adaptor 配置

在百胜开放平台应用审核通过之后，会根据申请应用中填的回调 URL 把对应的 appId、公私钥会自动同步到 Adaptor 中。也可以登录到 Adaptor 中到配置管理->外部系统配置中进行相关操作。



如果在列表中不存在的在开放平台申请的新的 appId 的，可以点击使用新增功能在新增页面选择已经在百胜开放平台审核通过的所有 appId 信息，填写相关信息之后点击确定即可。



如果在列表已经存在的，可以使用编辑功能进行编辑操作。

Adaptor 外部系统配置中，可以根据不同的平台、店铺配置不同的推送地址，可以按需使用，可以在新增界面或者编辑界面进行平台和店铺的选择。

4.1.2.6. 奇门 POS 配置

Adaptor 聚石塔环境与中台/ERP 对接，是通过阿里奇门 POS 场景进行对接的，需要进行奇门 POS 场景的授权操作。

4.1.2.6.1. 奇门后台设置（需要联系百胜伙伴进行配置）

1、申请开通奇门 POS 场景：

appkey 是使用店铺下单使用的 appkey 信息（23068014）

目标 key 使用的是百胜 POS 的 key（23625323）

服务地址填写 Adaptor 地址（聚石塔）：

客户分类	目标地址	备注
Adaptor SaaS 聚石塔	https://gatewaynew-adaptor.baison.net/jst/adaptor/api	
Adaptor 套装聚石塔	http://xxx.xxx.xxx.xxx:port/adaptor/api	IP 和端口根据项目实际部署情况进行填充



2、在目标 appkey 中同意

目标 appkey 是百胜的 POS key (23625323)

服务地址填写中台 gateway 地址：（端口根据实际情况调整）

中台/ERP	中台/ERP 地址	备注
E3+	http://xxx.xxx.xxx.xxx:port/gateway-qimen/e3plus-qimen-dispatch	IP 和端口根据项目实际部署情况进行填充
E3		
ME3		
自研中台	自研中台地址，由自研中台提供	



4.1.2.6.2. 客户方需要进行的操作（需要现场服务人员及客户方人员处理）

1、店铺订购 POS KEY (23625323)

订购地址：[https://fuwu.taobao.com/ser/detail.htm?spm=a1z13.pc_search_result.1234-fwlb.5.69815aca7M5pKm&service_code=FW_GOODS-](https://fuwu.taobao.com/ser/detail.htm?spm=a1z13.pc_search_result.1234-fwlb.5.69815aca7M5pKm&service_code=FW_GOODS-1000465752&tracelog=search&from_key=%E6%94%B6%E9%93%B6%E5%8F%B0)

[1000465752&tracelog=search&from_key=%E6%94%B6%E9%93%B6%E5%8F%B0](https://fuwu.taobao.com/ser/detail.htm?spm=a1z13.pc_search_result.1234-fwlb.5.69815aca7M5pKm&service_code=FW_GOODS-1000465752&tracelog=search&from_key=%E6%94%B6%E9%93%B6%E5%8F%B0)

注意：订购的时候，需要进行支付操作，支付的金额为 0

交易管理 / 服务后台订单
服务后台订单

订单号	店铺名称	收银台	默认收费项目	操作时间	商品信息	状态	操作
31-06			默认收费项目	2021-08-18 16:41:06			修改
31-0109	探店	百胜收银台	默认收费项目	2021-08-18 15:45:42	产品 0 实施 0 二次开发 0 服务 0 总价 0 首款 0 尾款 0	待付款	修改

2、订购完之后，需要百胜伙伴进行审核，审核完成之后，需要店铺授权给 POS APPKEY 获取 token 等信息，
appkey: 23625323

授权地址: <https://open.taobao.com/apitools/sessionPage.htm?spm=0.0.0.0.ivqDTi>

订购之后获取到对应的 token 信息（此 session 信息不需要提供）

注意：每个店铺都需要进行订购以及 token 获取（token 获取只是为了帮忙授权关系，不用提供 token 信息）

4.2. 中台/ERP 接入

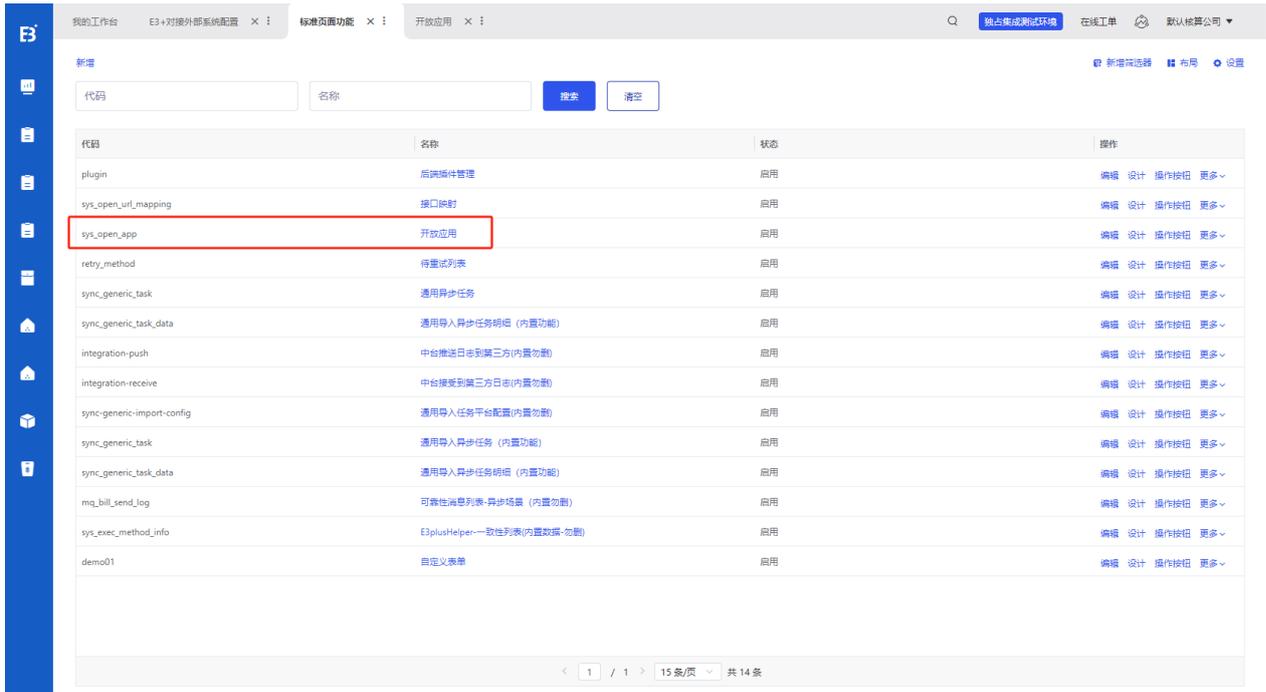
4.2.1. E3+企业中台

4.2.1.1. 开发中心添加 Adaptor 的 appId 公私钥

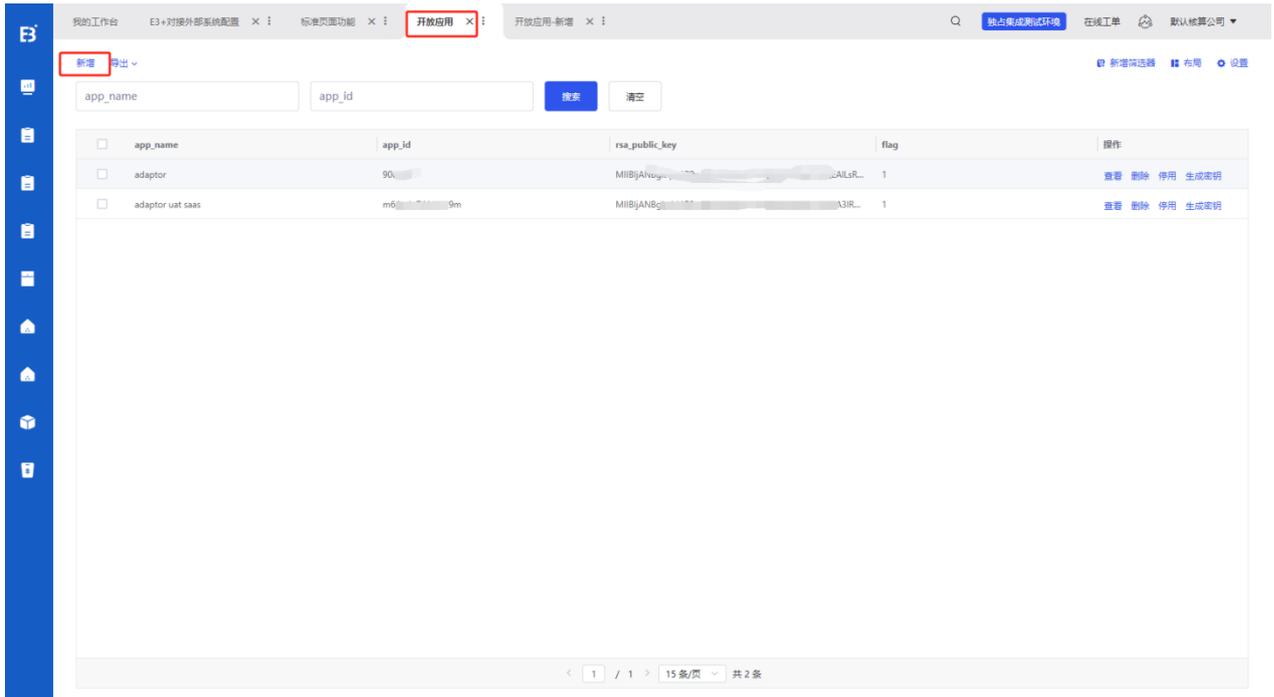
对应的场景：Adaptor 单据推送给中台的时候，中台会根据这里配置的 appId、公私钥进行签名校验
菜单位置：【开发中心】->【在线开发】->【标准页面功能】



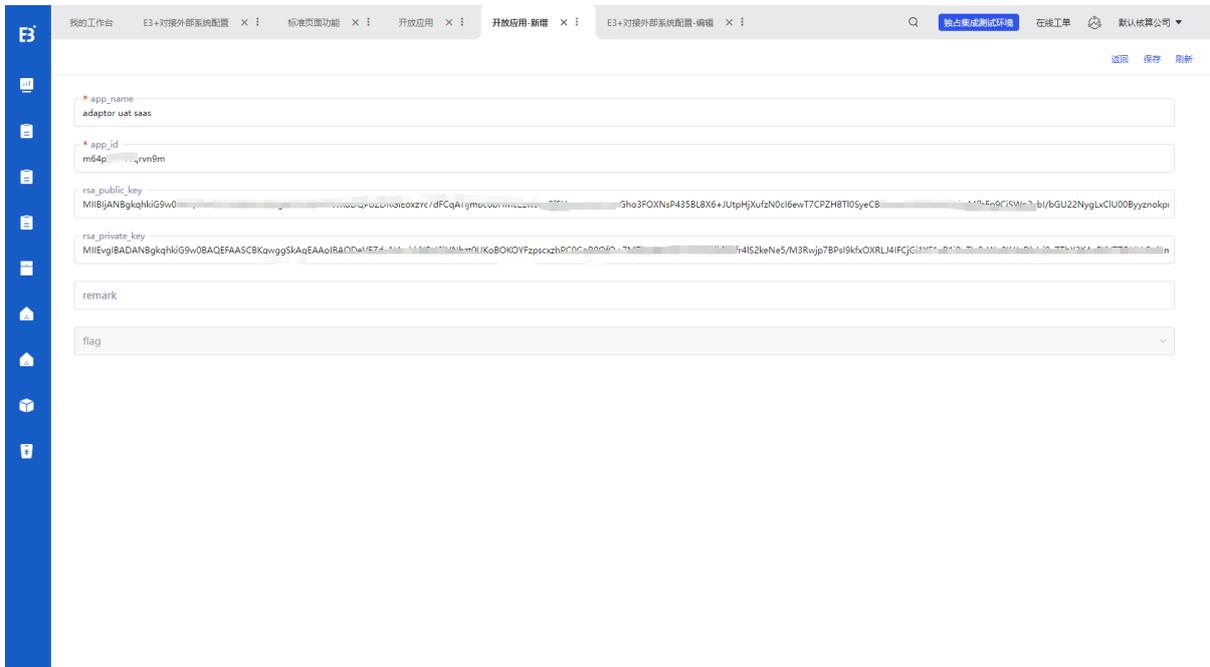
标准页面代码: 【sys_open_app】, 标准页面名称: 【开放应用】



在开放应用页面，进行新增



在新增页面，填写 app_name、app_id、rsa_public_key、rsa_private_key 信息之后进行保存操作。这里的 app_id、rsa_public_key、rsa_private_key 需要是在上一步中在百胜开放平台申请的应用对应的 appId、公钥信息。

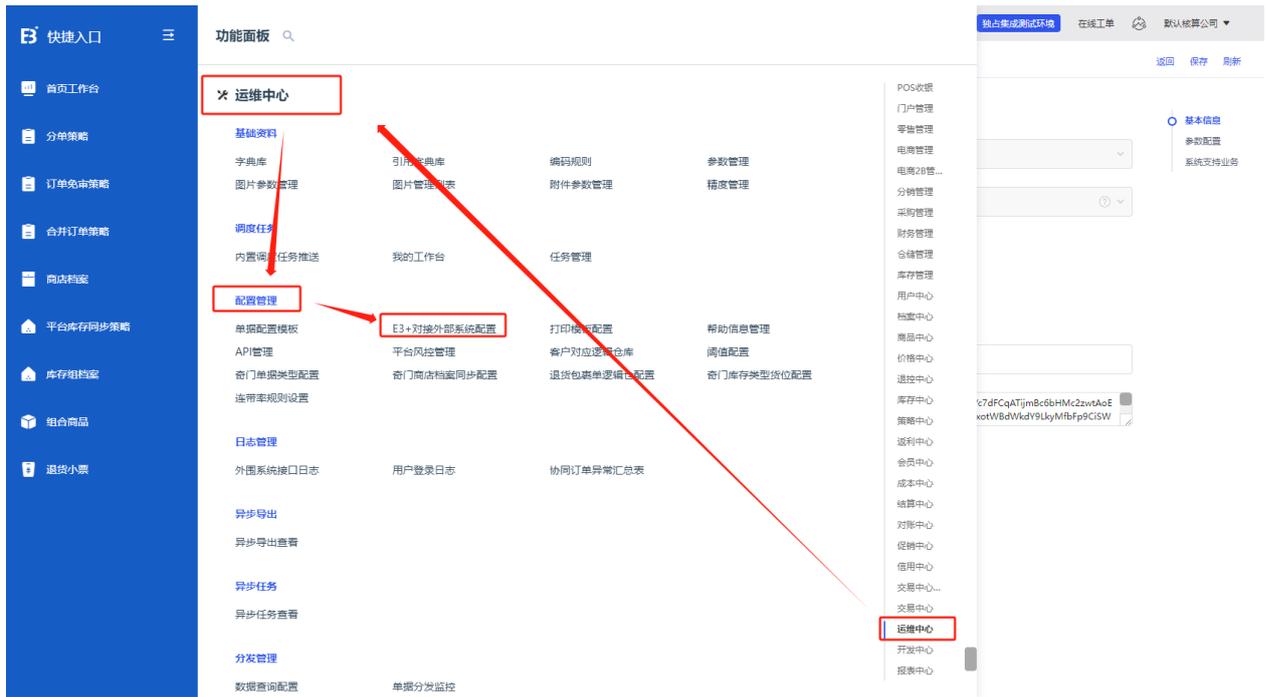


新增完成之后，在列表页面进行启用操作。

4.2.1.2. E3+对接外部系统配置增加 Adaptor 对接配置

对应的场景：

菜单位置：【运维中心】->【配置管理】->【E3+对接外部系统配置】



E3+对接外部系统配置列表：

代码	名称	对接场景	外部系统	状态	操作
KD100	快递100	快递	快递100	启用	查看 编辑 禁用
51TK	51tracking	快递	快递100	启用	查看 编辑 禁用
adaptor-tb	网上adaptor	电商适配网关配置	内部	启用	查看 编辑 禁用
FL_WMS	富勒奇门	WMS	富勒WMS	启用	查看 编辑 禁用
0111	OA	OA	泛微OA	初始	查看 启用 编辑 删除
adaptor-tx	adaptor线下	电商适配网关配置	内部	启用	查看 编辑 禁用
adaptor-jd	adaptor云端	电商适配网关配置	内部	启用	查看 编辑 禁用

新增中，填写必要的代码、名称、对接场景、外部系统及对接方式、所属平台之后进行保存

代码：E3+对接外部系统配置中唯一即可

名称：简要的易分辨的名称

对接场景：固定为【电商适配网关配置】

外部系统对接方式：

与 Adaptor 聚石塔环境对接，一般需要使用奇门对接方式

与其他 Adaptor 环境对接，一般使用内部对接方式
 所属平台：E3+在请求 Adaptor 的时候会根据此处配置的平台进行路由，需要正确填写

Adaptor 环境	适用平台	说明
聚石塔	淘系平台	淘系平台单据对接均需要通过 Adaptor 聚石塔环境
云鼎	京东系平台	京东系平台单据对接均需要通过 Adaptor 云鼎环境
塔下公用云	非淘系京东系平台	非淘系、京东系平台单据对接均需要通过 Adaptor 塔下环境 对于拼多多、抖音这两个平台的对接，目前基本是通过拼多多共享服务、抖店共享服务进行对接的，因此这两个平台适用于 Adaptor 塔下环境



添加成功之后，在编辑编辑页面填写对应的参数及系统支持业务信息

url（外网接口地址）：填写 Adaptor 提供的 api 地址

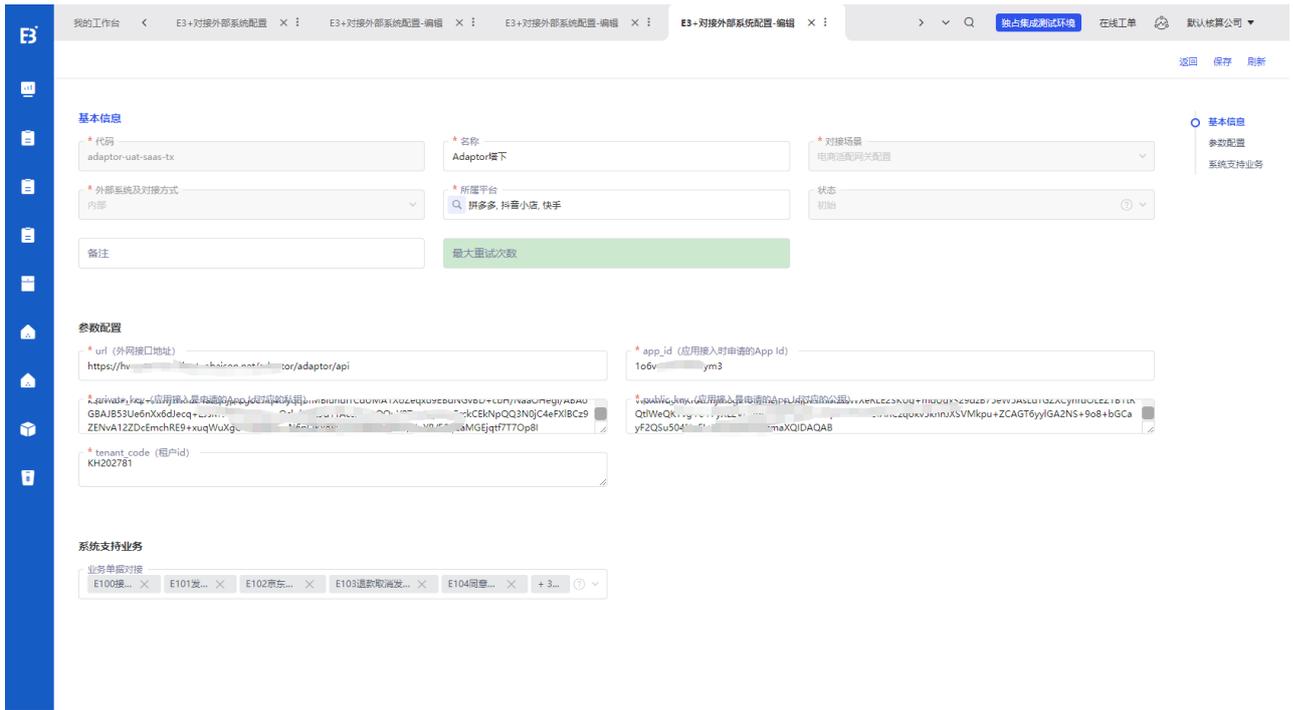
app_id（应用接入时申请的 App Id）：填写在开放平台中申请的 Adaptor 应用的 appId

private_key（应用接入时申请的 App Id 对应的私钥）：填写在开放平台中申请的 Adaptor 应用对应的私钥

public_key（应用接入时申请的 App Id 对应的公钥）：填写在开放平台中申请的 Adaptor 应用对应的公钥

tenant_code（租户 id）：Adaptor 当前租户编码

系统支持业务：可根据需要对接的场景进行选择，这些接口一般都与 E3+做了对接，可以根据客户业务情况进行选择



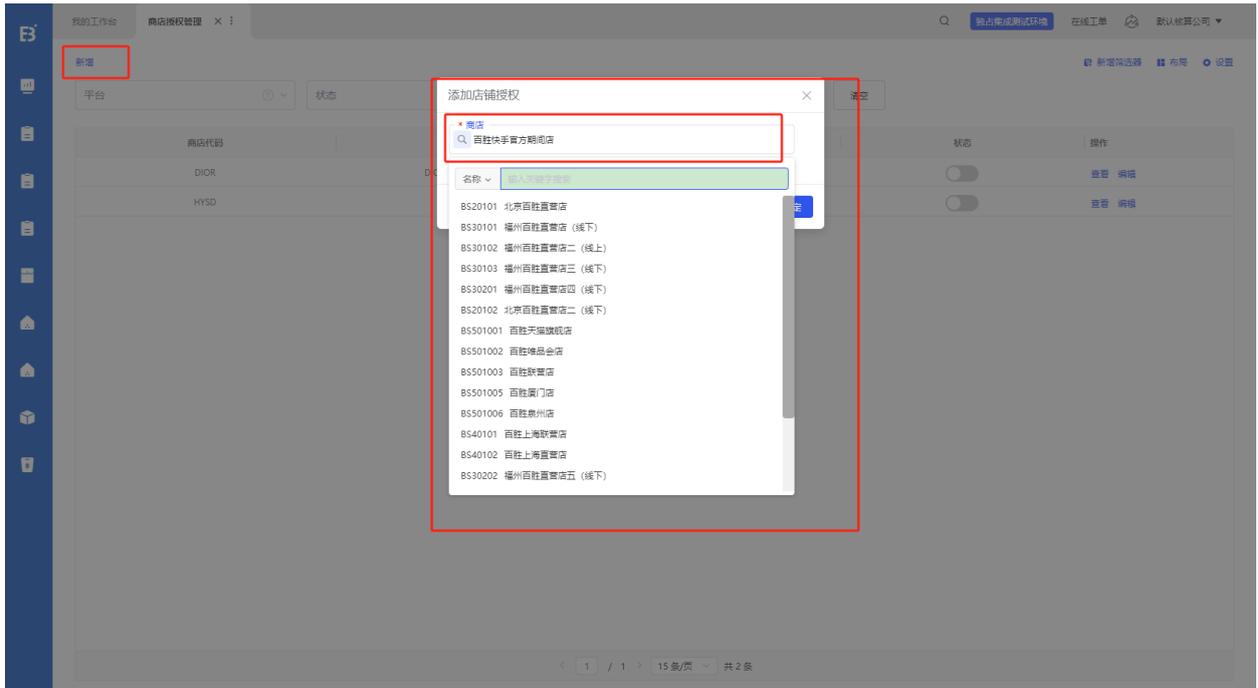
编辑完成之后，在列表界面启用即会生效

4.2.1.3. 商店授权管理

菜单位置：【电商管理】->【系统档案】->【商店授权管理】



商店授权页面进行新增操作
商店：选择需要添加授权的商店



商店添加完成之后，点击编辑，进入到商店授权详情界面，根据提示填写响应的参数
如果已经有 token 的，可以把 token 直接维护到商店授权中
获取 code:

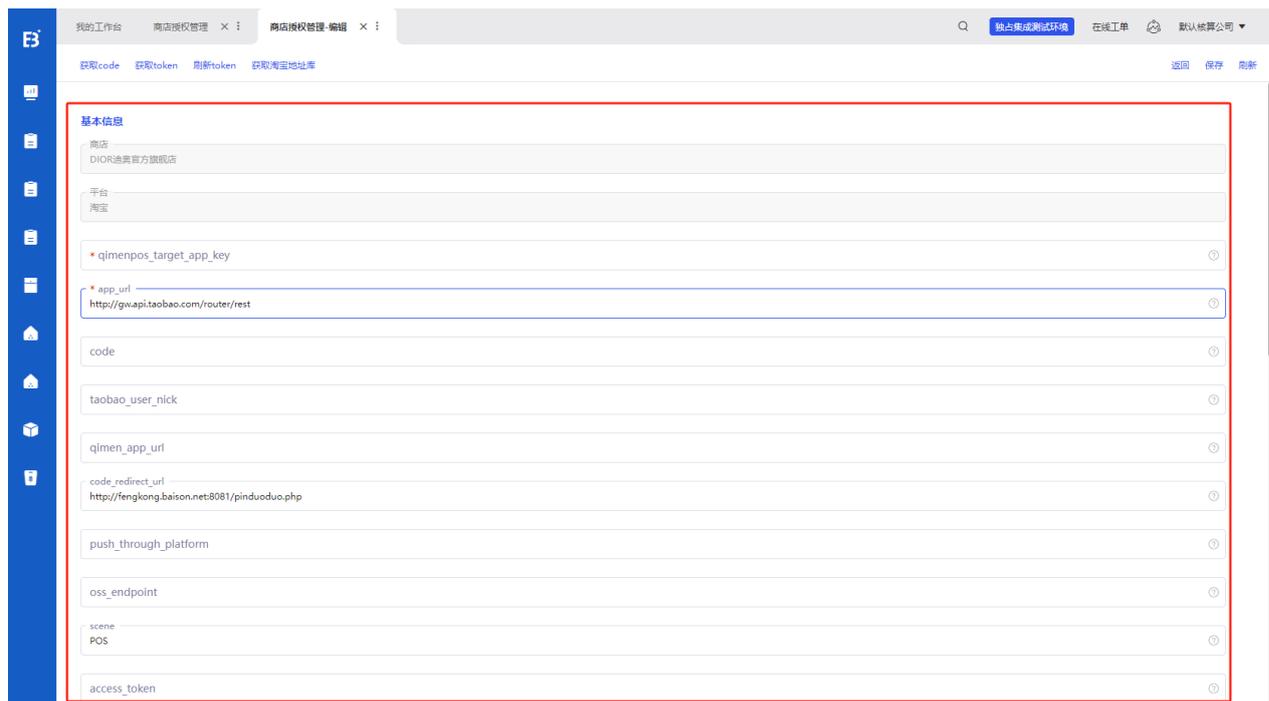
在进行店铺授权的时候，把 appkey 和 appsecret 维护进去之后，即可使用获取 code 功能，点击获取 code 会跳转到商家授权登录页面，需要商家进行授权登录，登录成功之后，在跳转的 URL 中会返回平台的 code，获取到 code 之后，把 code 填充到商店授权中

获取 token:

把 code 填入之后，点击获取 token，后端根据 appkey、appsecret 以及上一步中填入的 code，到平台获取 token，获取 token 之后会把商店授权中的 token 及 refreshtoken 进行填充

刷新 token:

token 过期之后，如果已经有维护正确的 refreshtoken 的话，可以点击刷新 token 进行 token 有效期的刷新



完成以上步骤，Adaptor 和 E3+就全部联通了，联通之后就可以进行正常单据的推送和状态的回写了！

4.2.2. E3 全渠道中台

知识库文档地址：<http://bsconfluence.baison.net:40506/pages/viewpage.action?pageId=137759412>

4.2.2.1. 增值接口管理

场景：E3 项目第一次对接 Adaptor，需要获取增值接口信息

菜单位置：【系统】->【授权管理】->【增值接口列表】

4.2.2.2. Adaptor 参数设置

菜单位置：【配置管理】->【adaptor 参数设置】

根据页面提示进行填写即可

工作台 adaptor参数设置

对接adaptor参数设置

appId:	<input type="text"/>	adaptor平台上的appId
tenantCode:	<input type="text"/>	adaptor平台上的租户编码
私钥:	<div style="border: 1px solid #ccc; height: 60px;"></div>	adaptor平台上的私钥
公钥:	<div style="border: 1px solid #ccc; height: 60px;"></div>	adaptor平台上的公钥
公有云URL:	<input type="text"/>	对接adaptor公有云的接口地址,http://gateway.adaptor.baison.net/adaptor/adaptor/api,默认可不填
聚石塔URL:	<input type="text"/>	对接adaptor聚石塔的接口地址,http://jst-adaptor-saas.baison.net/adaptor/api,默认可不填
云鼎URL:	<input type="text"/>	对接adaptor云鼎的接口地址,http://116.196.82.170:30005/adaptor/api,默认可不填
短信通URL:	<input type="text"/>	对接adaptor短信通的接口地址

[保存](#)

4.2.2.3. 商店档案管理

菜单位置：【档案】->【商店档案】

进行店铺新建即可，默认不填写网店参数，但是维护外部商店编码，店铺创建成功之后会调用 Adaptor 接口进行店铺创建，店铺参数等都在 Adaptor 中进行维护

查看京东自营... 商店档案 编辑OH微盟智... 零售门店列表 添加商店

店

代码	shopify *	名称	shopify *
店铺性质	直营店 *	店铺类型	shopify * 请输入搜索关键字 x
渠道	总部 * 请输入搜索关键字 x	类别	B2C *
区域	北京 * 请输入搜索关键字 x	业务员	
结算性质	<input type="checkbox"/> 销货结算 <input type="checkbox"/> 零售结算	客户等级	A
仓库	总仓 * 请输入搜索关键字 x	客户	线上客户 * 请输入搜索关键字 x
退货仓库	实际发货仓库 *	O2O订单默认退货仓库	请选择 请输入搜索关键字 x
换货追单商品默认发货仓库	请选择 请输入搜索关键字 x		
备注		线上	<input type="checkbox"/>
外部商店编码	shopify	对接ERP商店编码	
销售平台扣点比例	(0~100)	平台付款货币	请选择

4.2.2.4. 网店快递方式映射

店铺授权信息维护完成之后，在 E3 中点击获取网店快递方式，与 E3 中的快递方式进行映射匹配

4.2.2.5. 需要开启的计划任务

每个平台均有不同的计划任务需要开启，详情参考 E3 知识库文档。

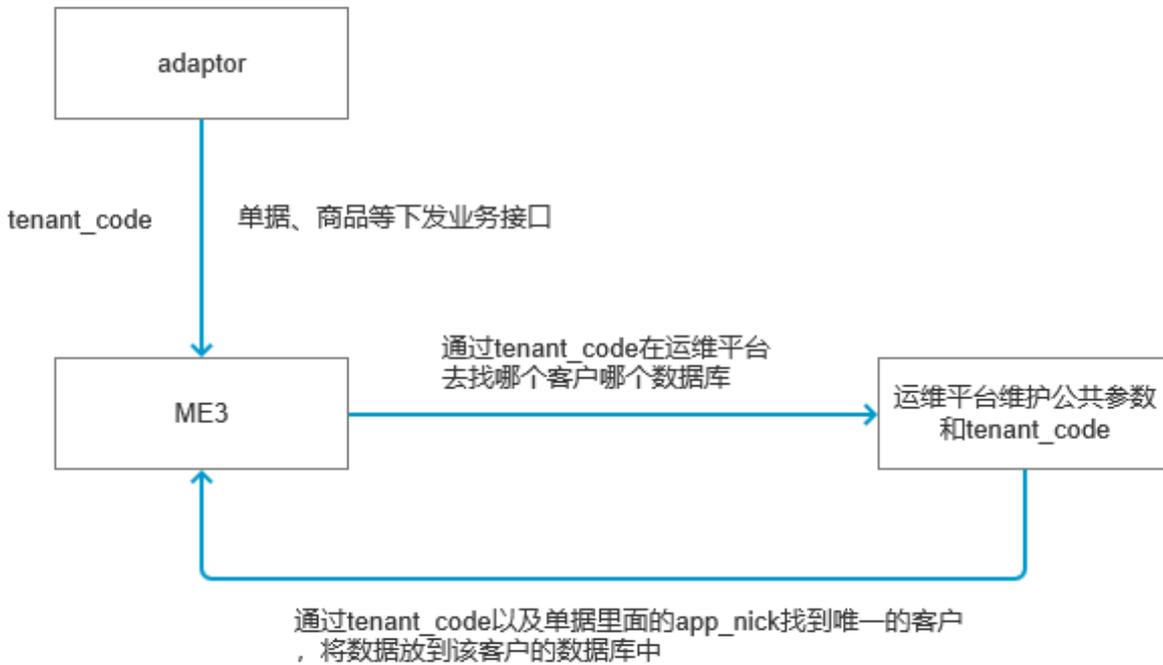
4.2.3. ME3 全渠道云中台

4.2.3.1. 运维平台配置

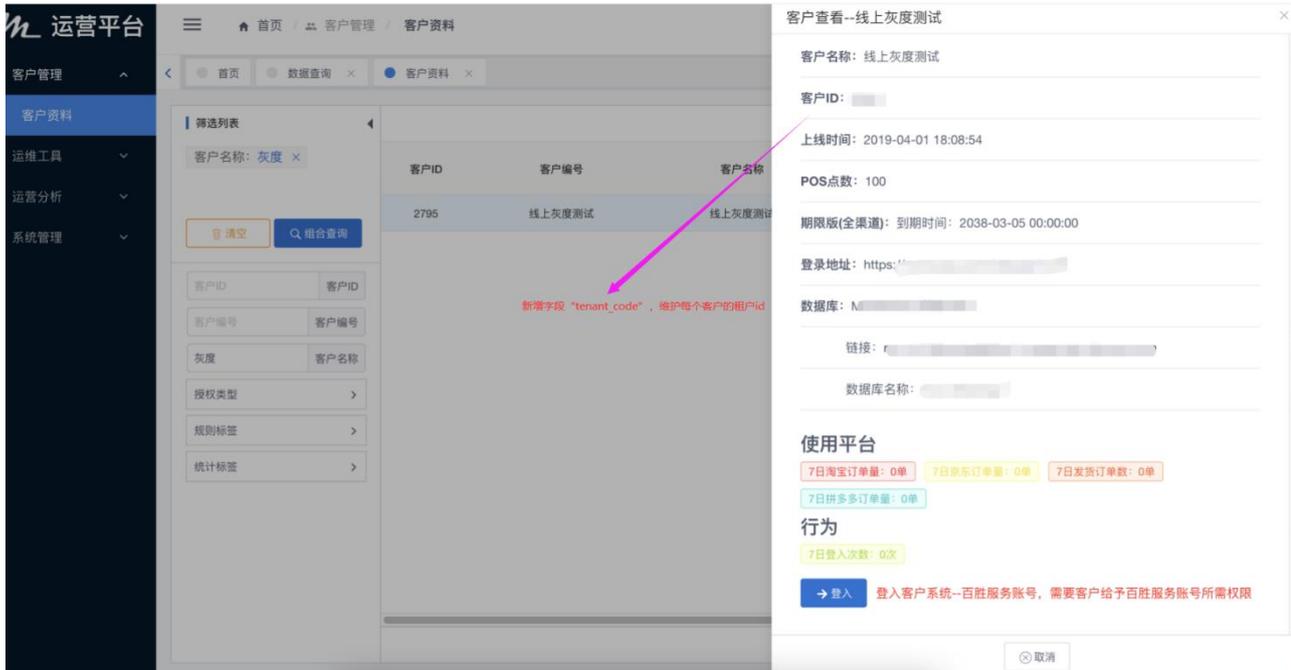
1、运维平台新增【Adaptor 平台配置】维护公共参数：

- 1) app_id: adaptor 分配给外系统的应用 ID (adaptor 获取)
- 2) api_url: adaptor 的外网地址，调用 adaptor 服务必填
- 3) publicKeyPem(公钥): adaptor 生成固定值，需要从 adaptor 获取
- 4) privateKeyPem(私钥): adaptor 生成固定值，需要从 adaptor 获取
- 5) 平台类型: 支持 adaptor 对接了的平台（目前 ME3 仅淘宝、淘分销对接，其他平台后续待完善）

app_id	<input type="text"/>	*	adaptor分配给外系统的应用ID			
api_url	<input type="text"/>	*	adaptor的url接口			
publicKeyPem(公钥)	<input type="text"/>	*	adaptor生成固定值，需要从adaptor获取			
privateKeyPem(私钥)	<input type="text"/>	*	adaptor生成固定值，需要从adaptor获取			
平台类型	<input checked="" type="checkbox"/> 淘宝	<input type="checkbox"/> 京东	<input type="checkbox"/> 拼多多	<input type="checkbox"/> 抖音小店	<input type="checkbox"/> 爱库存	<input type="checkbox"/> 唯品会JIT
	<input type="checkbox"/> 有赞	<input type="checkbox"/> 亚马逊	<input type="checkbox"/> 当当网	<input type="checkbox"/> 苏宁易购	<input type="checkbox"/> 阿里巴巴	<input type="checkbox"/> 小红书
	<input type="checkbox"/> 快手小店	<input type="checkbox"/> 好衣库	<input type="checkbox"/> 微店	<input type="checkbox"/> 天猫超市	<input type="checkbox"/> 网易严选	<input type="checkbox"/> 微信小店



6) 在运维平台的【客户参数】里面新增字段 tenant_code: 使用 SAS 部署的 adaptor 需要用到此值, 此值由 adaptor 针对每个客户分配, 每个客户绑定一个租户 id, 后面对接传输数据到 ME3 的各个接口也都会传此字段;



4.2.3.2. ME3 配置

店铺创建&授权

ME3和adaptor交互-店铺创建和授权



1) 在全渠道中台 ERP 业务系统中新增店铺传到 ME3，ME3 的【网点档案设置】中二次编辑选择店铺平台类型，当选择淘宝/淘分销类型店铺保存后调用 Adaptor 接口“E9001-shop 商店新增 e3plus.adaptor.shop.add”，在 Adaptor 中创建 ME3 对应的商店；

2) ME3 的【网点档案设置】中店铺点击授权，调用 Adaptor 接口“E9008-shop 商店获取 session(跳转地址)e3plus.adaptor.shop.auth.token.get”，Adaptor 会根据不同的平台进行 session/token 的获取，获取之后会保存到 Adaptor 的数据库中并返回给 ME3 成功；**如果授权成功**则 Adaptor 返回给 ME3 成功并且再触发 ME3 调用 Adaptor 接口“E9002-shop 商店详情 e3plus.adaptor.shop.get”接口获取到最新的 session 并存储在 ME3；**如果授权失败**则弹出弹框：

点击右上角的 X 关闭弹出框不做任何操作和接口调用；点击【重新授权】则再次调用上一步的接口重新获取 session；



4) ME3 线上类型的店铺状态点击启用之后触发调用 Adaptor 接口“E9004-shop 商店启用 e3plus.adaptor.shop.enable”，根据商店 code 启用商店；

5) ME3 线上类型的店铺状态点击禁用之后触发调用调用 Adaptor 接口“E9005-shop 商店禁用 e3plus.adaptor.shop.disable”，根据商店 code 禁用商店；

注意：ERP 的店铺启用和禁用不影响 ME3 店铺的启用和禁用；

4.2.4. 企业自建中台/ERP

根据企业自建中台/ERP 要求进行配置

4.3. 商城/小程序接入

4.3.1. 客制化开发

详情参考 2.3.2.客制化开发

4.3.2. 商城自助接入 API

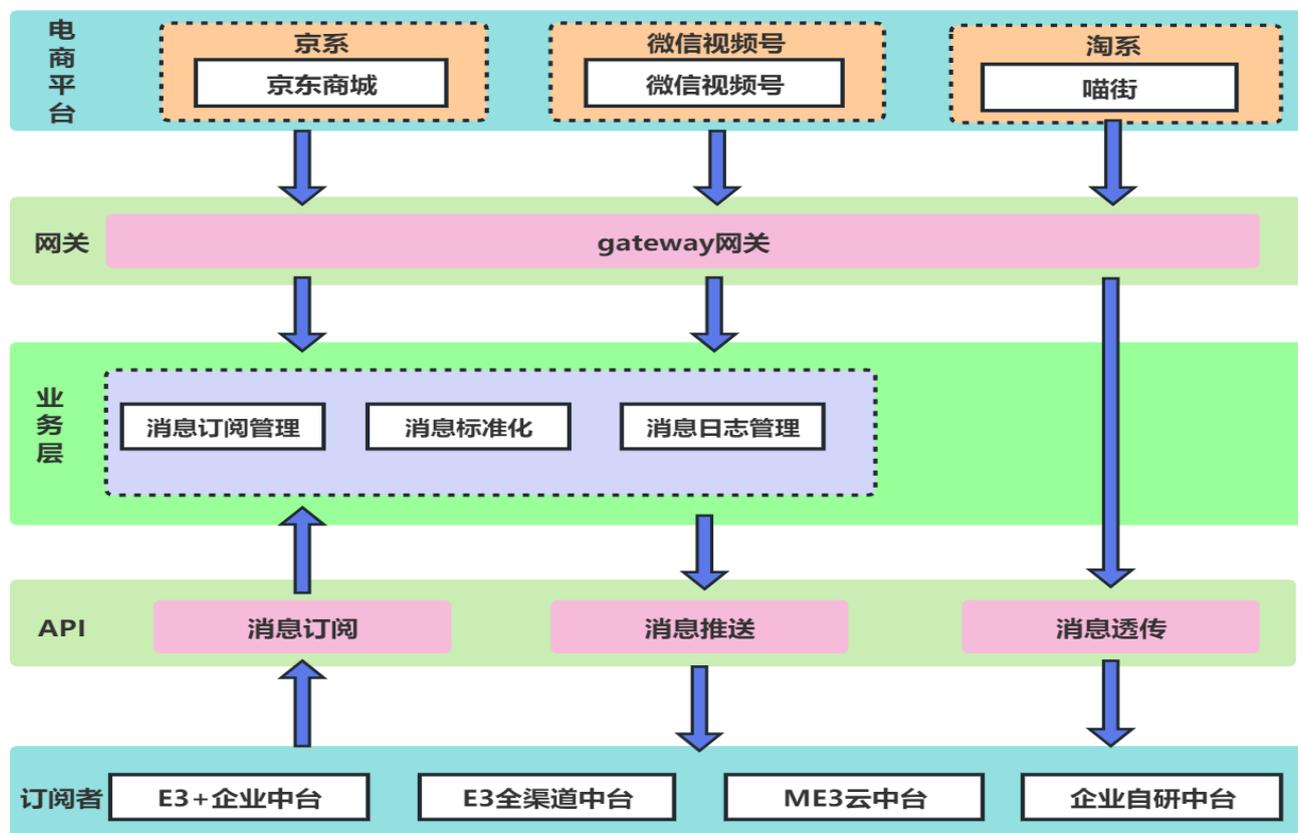
详情参考 2.3.3.商城自助接入 API

五、 消息服务

电商平台消息服务是为了帮助电商平台实现高效的单据推送，提升单据流转效率，同时减少列表接口的请求量。对于应用服务，消耗最大的操作就是列表查询，电商平台通过消息服务，有效的降低对服务压力比较大的列表查询操作。下游中台/ERP 等系统通过订阅平台消息服务，使得平台可以通过消息服务及时的推送平台的单据，下游中台/ERP 系统不需要多次轮询访问列表查询 API，提升数据传输效率。

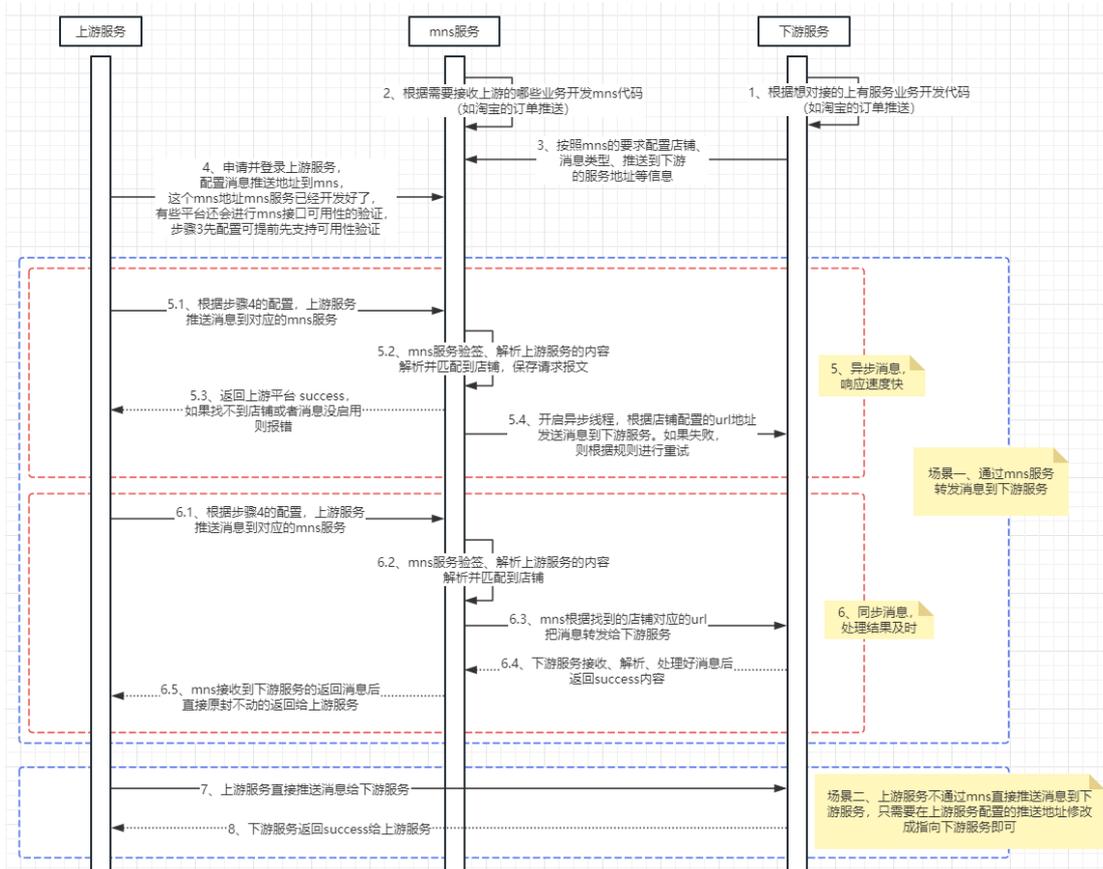
Adaptor 消息服务作为 E3 消息服务的补充，实现了 E3 消息服务未对接的平台，并接口标准沿用了 E3 消息服务的接口标准，可以与 E3 消息服务接口兼容，降低下游系统对接成本。

5.1. 架构设计



5.2. 交互说明

消息服务交互图:



- 上游服务：电商平台的消息服务，比如淘宝、京东、微信视频号小店等平台消息推送服务。
- Mns 服务：Adaptor 消息服务。
- 下游服务：电商业务系统，即处理电商订单核心履约的业务系统，比如 Adaptor 全网通、E3+企业中台、E3 全渠道中台、ME3 全渠道云中台及企业自研中台/ERP 服务。

5.3. 对接平台

消息服务已经对接了微信视频号小店、京东商城、闲鱼等 9 个平台。

平台	平台编码(platform)	平台编号(lylx)	消息类型	描述
美团团好货	meituantuanhaohuo	167	同步透传	
海拍客	haipaike	1014	同步透传	
喵街	miaojie	1013	同步透传	
猫享自营	maoxiangziying	1016	同步透传	
VTN	vtn	1017	同步透传	
Shoptline	shoptline	1018	同步透传	
微信视频号小店	wxsphxd		异步订阅	
京东商城	jingdong		异步订阅	
闲鱼	xianyu		异步订阅	

Adaptor 消息服务设计的时候为了与 E3 消息服务做到兼容，增加了 E3 消息服务使用的平台编号 (lylx) 字段，lylx 类型为 E3 全渠道中台平台的数字 id。

同步透传消息为平台不支持异步处理的消息，需要业务中台系统即时的返回成功失败的强一致型消息；异步订阅消息为异步消息类型，允许 mns 消息服务在接收到消息之后可以异步推送业务中台/ERP 系统。

5.4. Adaptor 平台消息订阅配置

菜单位置：【配置管理】->【平台消息订阅配置】

适用场景：对于微信视频号、闲鱼、京东商城等支持异步消息推送的平台，在此菜单进行配置

- 开启消息订阅：总开关，开启的情况下配置有效，如果选择关闭，则 Adaptor 会把所有订阅类型的消息进行关闭。
- 平台类型：选择店铺所在的平台，会自动把系统中对应平台的店铺信息带过来。
- 店铺名称：选择需要对接消息服务的店铺。
- 开启批量推送：开启之后，订阅消息 mns 消息服务会批量推送给 Adaptor，建议使用批量处理减少 IO，提升系统效率。
- 公司名称：Adaptor SaaS 一般取当前登录的租户名称，Adaptor 套装一般取喷码信息；支持自定义维护。
- 授权 IP 地址：可直接使用当前系统（Adaptor）的 IP 地址
- 数据转发地址：当前应用（Adaptor）的消息推送后端接口地址，承接 mns 消息服务推送的消息。可以通过点击【使用当前主机地址】自动获取到当前系统的后端地址，也可以通过开发、交付、运维等人员配置后端服务的地址。

注意：每个平台需执行一次保存操作，否则会导致已经选择的平台的店铺和消息类型信息会丢失。

5.5. Adaptor 平台接口路由

菜单位置：【配置管理】->【平台接口路由】

适用场景：对于猫享自营、美团团好货等平台消息要求强一致的，在此菜单进行配置

- 数据转发地址：当前应用（Adaptor）的消息推送后端接口地址，承接 mns 消息服务推送的消息。可以通过点击【使用当前主机地址】自动获取到当前系统的后端地址，也可以通过开发、交付、运维等人员配置后端服务的地址。
- 公司名称：Adaptor SaaS 一般取当前登录的租户名称，Adaptor 套装一般取喷码信息；支持自定义维护。
- 平台类型：选择店铺所在的平台，会自动把系统中对应平台的店铺信息带过来。
- 店铺名称：选择需要对接消息服务的店铺。
- 消息类型：选择需要对接消息服务的消息类别。

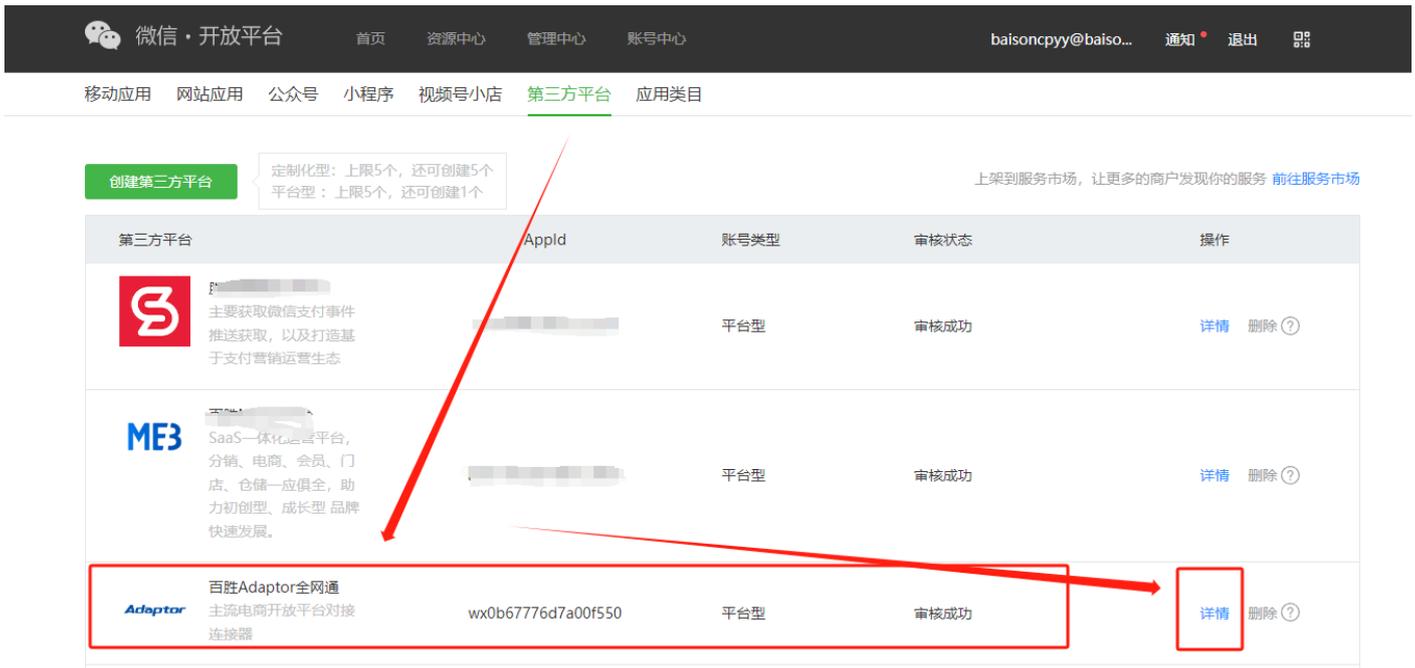
注意：每个平台需执行一次保存操作，否则会导致已经选择的平台的店铺和消息类型信息会丢失。

5.6. 平台后台配置消息推送地址

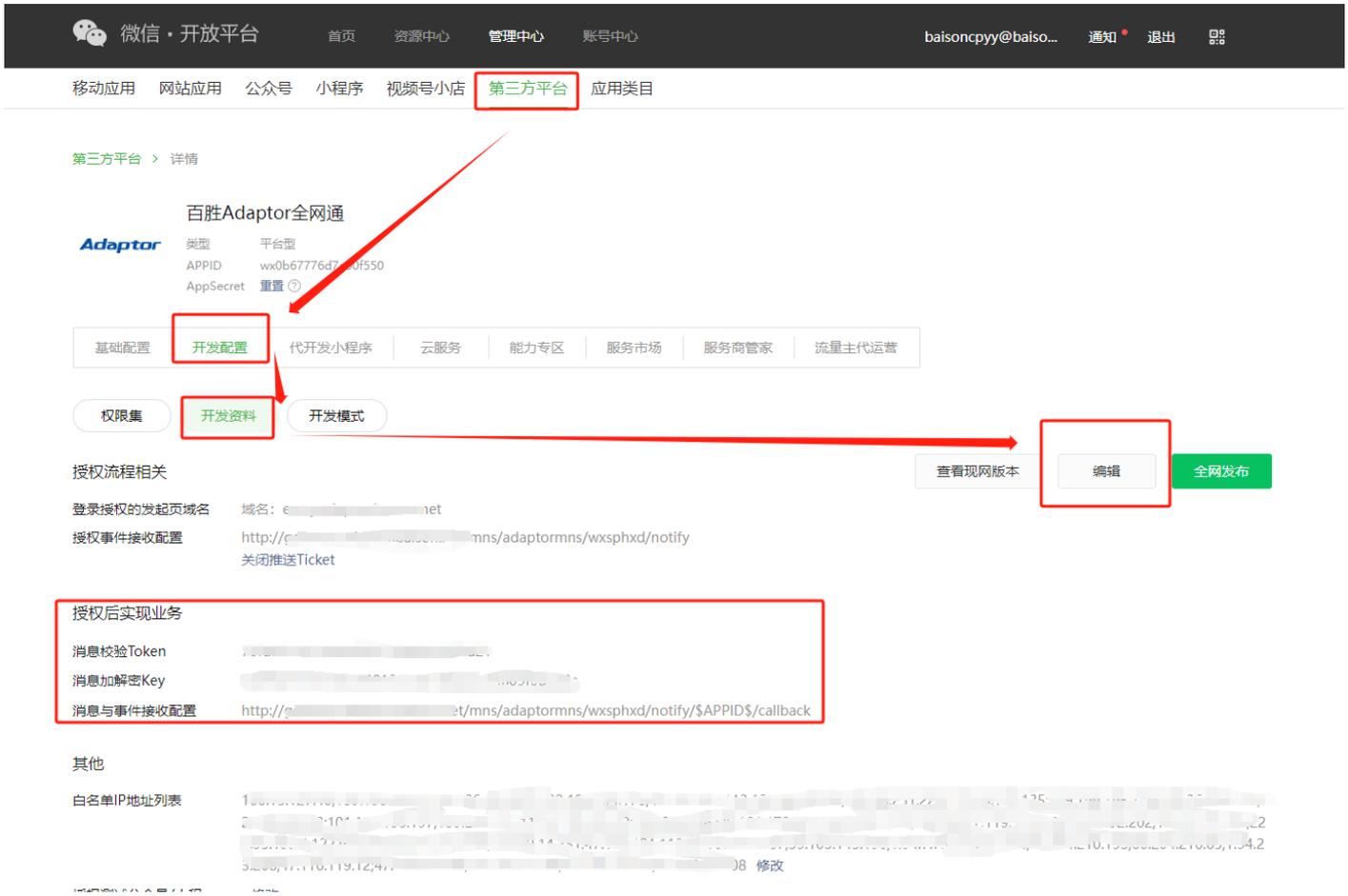
5.6.1. 微信视频号小店

- 1、登录微信开放平台后台，后台地址：<https://open.weixin.qq.com/>

找到需要进行配置的应用：



2、在【开发配置】->【开发资料】部分可以看到配置的消息推送地址，需要配置或者调整的，点击【编辑】进入到编辑页面进行配置



3、在【消息与事件接收】部分进行推送地址和安全校验配置

授权事件接收配置：用于接收消息或者事件的推送，比如授权事件的推送等

消息与事件接收配置：用于接收平台消息的推送，单据消息等的推送均是通过此地址进行推送的

消息校验 Token：用于接收到推送的消息之后，对签名等进行校验

消息加解密 Key：用于消息加密推送过程中平台的解密及消息接收之后的解密



第三方平台 > 编辑开发资料

确保已经仔细阅读申请资料说明，再填写开发资料。

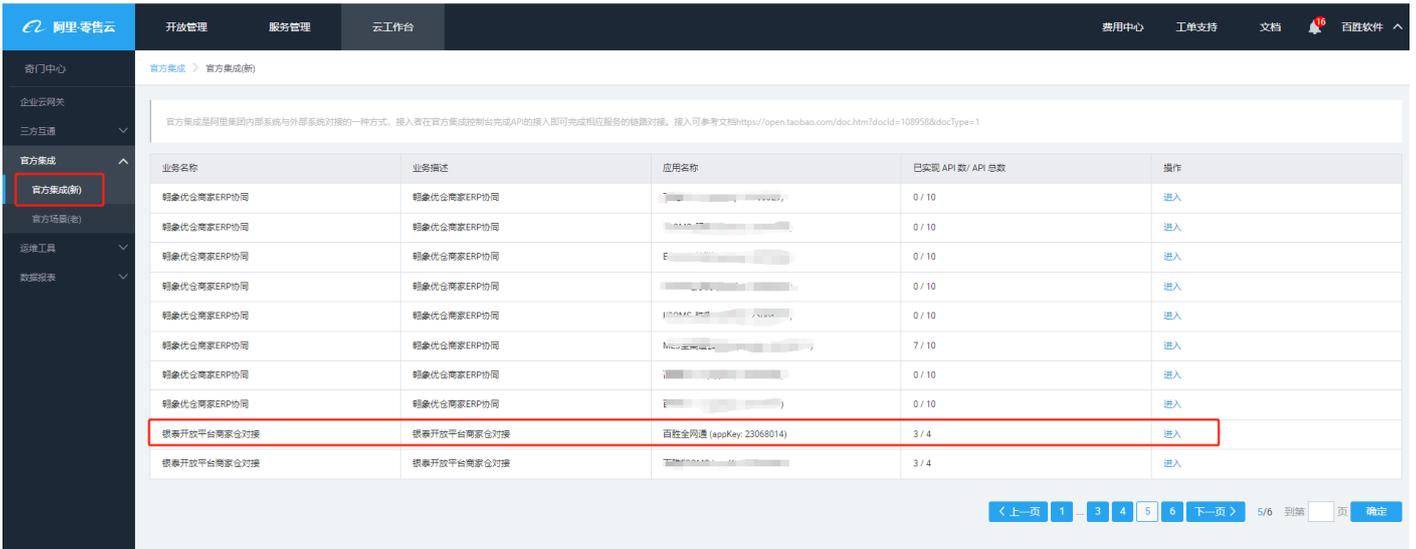
修改开发资料，仅对测试公众号/小程序生效，届时提交“全网发布”并成功后，才可影响现网；

消息与事件接收	
授权事件接收配置	<p>http:// [domain] /mns/adaptormns/v 56</p> <p>用于接收平台推送给第三方平台账号的消息与事件，如授权事件通知、component_verify_ticket等。 注意：该URL的一级域名需与“消息与事件接收配置”的一级域名一致 云开发模式下第三方平台接收事件通知以及获取调用凭据，请查看详情。</p>
消息与事件接收配置	<p>http:// [domain] /mns/adaptormns/v 73</p> <p>用于代授权的公众号或小程序的接收平台推送的消息与事件，该参数按规则填写（需包含\$APPID\$，如www.abc.com/\$APPID\$/callback），实际接收消息时\$APPID\$将被替换为公众号或小程序AppId。 注意：该URL的一级域名需与“授权事件接收配置”的一级域名一致 云开发模式下第三方平台接收消息的方式，查看详情。</p>
消息校验Token	<p>[token] 32</p> <p>开发者在代替公众号或小程序接收到消息时，用此Token来校验消息。</p>
消息加解密Key	<p>[key] 43</p> <p>在代替公众号或小程序收发消息过程中使用。必须是长度为43位的字符串，只能是字母和数字。</p>

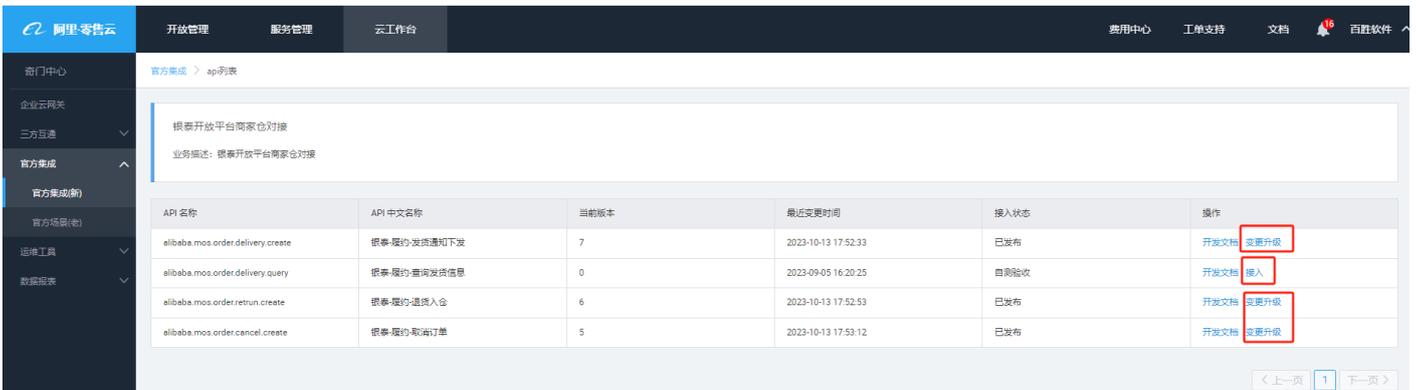
5.6.2. 喵街

喵街需要在奇门云工作台后台进行配置，登录地址：<https://qimen.taobao.com/>

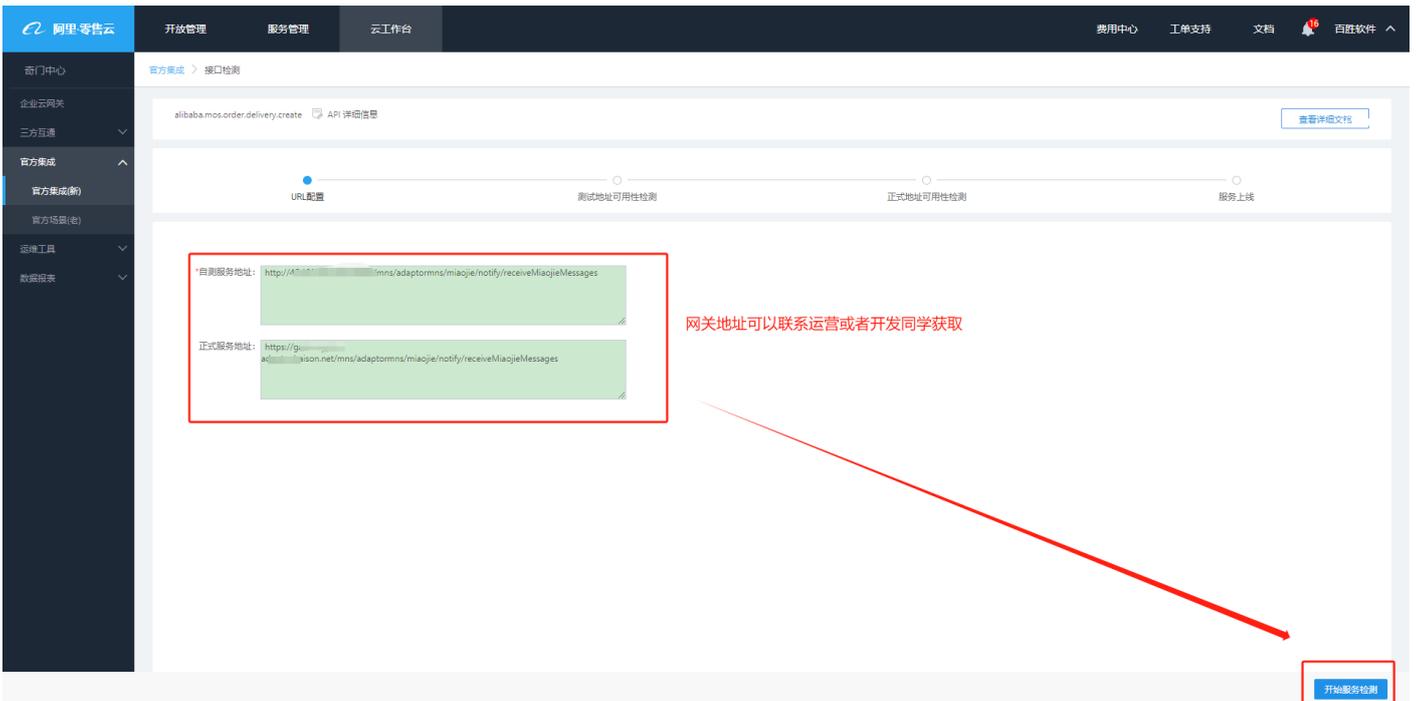
1、在【官方集成】-【官方集成(新)】菜单中，找到业务名称为：银泰开放平台商家仓对接的项目，进入到 api 列表页面；



2、在 api 列表页面，选择需要对接的 API 名称项目，点击接入到 URL 配置页面



3、填写自测地址及生产地址之后，点击【开始服务检测】，进行服务的检测，后面就是下一步下一步即可。



5.7. 消息服务管理功能

5.7.1. 消息服务 appId 管理

消息服务 appId 管理，是给接入 mns 消息服务授权进行管理的界面，可以根据需要进行新增，目前一般根据产品维度进行授权的。

app 名称：消息服务颁发给下游的应用名称

appKey：消息服务颁发给下游的应用信息

appSecret：消息服务颁发给下游的密钥信息

app名称	appid	appKey	appSecret	启用状态	创建时间	更新时间	操作
塔下adaptor (公共)	0...	adaptorMns	查看	启用	2023-08-03 10:26:11	2023-08-03 10:26:11	编辑 删除 禁用
ME3	0e...	adaptorMnsME3	查看	启用	2023-08-03 10:26:11	2023-08-03 10:26:11	编辑 删除 禁用
E3	0...	adaptorMnsE3	查看	启用	2023-08-03 10:26:11	2023-08-03 10:26:11	编辑 删除 禁用

5.7.2. 目标服务信息

目标消息服务可查看当前已经订阅消息服务的租户信息，以及使用的授权 appKey 等信息。

服务名称：Adaptor 中接入时候填写的公司名称，Adaptor SaaS 一般取当前登录的租户名称，Adaptor 套装一般取喷码信息；支持自定义维护。

appKey：当前接入使用的 mns 消息服务下发的 appKey 信息。

授权喷码：当前接入系统带过来的授权喷码信息。

服务Id	服务名称	服务描述	appKey	私钥	授权喷码	操作
1655507622605451265			adaptorMns	查看私钥		编辑 删除 禁用 配置平台推送地址
1646394650329161729			adaptorMns	查看私钥		编辑 删除 禁用 配置平台推送地址
1641649996019810306			adaptorMns	查看私钥		编辑 删除 禁用 配置平台推送地址
5659865326598658306			adaptorMns	查看私钥		编辑 删除 禁用 配置平台推送地址
1650076324728958978			adaptorMnsE3	查看私钥		编辑 删除 禁用 配置平台推送地址
1650076084483420162			adaptorMnsME3	查看私钥		编辑 删除 禁用 配置平台推送地址
1648637476064534530			adaptorMns	查看私钥		编辑 删除 禁用 配置平台推送地址
1646778232306475010			adaptorMns	查看私钥		编辑 删除 禁用 配置平台推送地址
1638366720469610498			adaptorMns	查看私钥		编辑 删除 禁用 配置平台推送地址

5.7.3. 店铺对照管理

店铺对照管理中可以查看已经订阅消息服务所有的店铺信息

店铺名称: Adaptor 中接入时候选择的店铺名称

平台编码: 店铺所属的平台编码

平台名称: 店铺所属的平台名称

目标服务名称: Adaptor 中接入时候填写的公司名称, Adaptor SaaS 一般取当前登录的租户名称, Adaptor 套装一般取喷码信息; 支持自定义维护

店铺名称	平台编码	平台名称	平台唯一码	服务编码	目标消息服务Id	目标消息服务名称	操作
视频号小店	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	164639464037705730	总部	编辑 禁用 删除
视频号小店(京东)	jingdong	京东		3	1712309858862182402		编辑 禁用 删除
电器京东自营旗舰店	jingdong	京东		79	1712309858862182402		编辑 禁用 删除
视频号小店官方旗舰店	wxsphwd	微信视频号小店	wx21b1d1e1d42	1717b0	1712309858862182402		编辑 禁用 删除
视频号小店官方旗舰店	wxsphwd	微信视频号小店	1		1712309858862182402		编辑 禁用 删除
视频号小店官方旗舰店	wxsphwd	微信视频号小店	wx21b1d1e1d42	0	5659865326598651234	北方总部	编辑 禁用 删除
视频号小店回测测试	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	5659865326598651234		编辑 禁用 删除
视频号小店	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	1646394668562365230		编辑 禁用 删除
视频号小店官方旗舰店	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	1646394668562365230		编辑 禁用 删除
视频号小店	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	1638665689562659534		编辑 禁用 删除
视频号旗舰店(零售)	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	1638716731540201474		编辑 禁用 删除
视频号黄金旗舰店	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	1638716731540201474		编辑 禁用 删除
视频号官方旗舰店(零售)	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	1638716731540201474		编辑 禁用 删除
视频号五号店	wxsphwd	微信视频号小店	wx21b1d1e1d42	wx21b1d1e1d42	1642769112147443713		编辑 禁用 删除

5.7.4. 店铺消息推送 topic 订阅管理

店铺订阅消息类型列表

店铺名称: Adaptor 中接入时候选择的店铺名称

平台编码: 店铺所属的平台编码

平台名称: 店铺所属的平台名称

目标服务名称: Adaptor 中接入时候填写的公司名称, Adaptor SaaS 一般取当前登录的租户名称, Adaptor 套装一般取喷码信息; 支持自定义维护

推送 topic: Adaptor 中接入时候选择的消息类型

平台编码	平台名称	目标信息服务名称	店铺名称	店铺id	目标信息服务id	推送topic	操作
177496003899394	wsphxd	微信视频号小店	微信 [店铺名称] 官方旗舰店	1703591223629000706	164063822463265026	订单更新	编辑 删除
187356163764225	jingdong	京东	[店铺名称] 官方旗舰店(京东)	1713746322439352321	1712309858862182402	退款下载	编辑 删除
187356570611713	jingdong	京东	[店铺名称] 官方旗舰店	1713746322489683969	1712309858862182402	订单更新	编辑 删除
*46322791673858	jingdong	京东	[店铺名称] 官方旗舰店	1713746322489683969	1712309858862182402	订单完成	编辑 删除
187356230873090	jingdong	京东	[店铺名称] (京东)	1713746322439352321	1712309858862182402	订单更新	编辑 删除
187356499308546	jingdong	京东	[店铺名称] 官方旗舰店	1713746322489683969	1712309858862182402	退款下载	编辑 删除
*46322678427650	jingdong	京东	[店铺名称] 官方旗舰店(京东)	1713746322439352321	1712309858862182402	订单完成	编辑 删除
187356100849666	jingdong	京东	[店铺名称] (京东)	1713746322439352321	1712309858862182402	订单下载	编辑 删除
187356436393986	jingdong	京东	[店铺名称] 旗舰店	1713746322489683969	1712309858862182402	订单下载	编辑 删除
*46322732953602	jingdong	京东	[店铺名称] 官方旗舰店	1713746322489683969	1712309858862182402	修改地址/备注	编辑 删除
*46322619707394	jingdong	京东	[店铺名称] (京东)	1713746322439352321	1712309858862182402	修改地址/备注	编辑 删除
*46120106127362	wsphxd	微信视频号小店	[店铺名称]	1713746119921577986	1712309858862182402	商品更新	编辑 删除
*46120160653314	wsphxd	微信视频号小店	[店铺名称] 司 [店铺名称] 网店	1713746119921577986	1712309858862182402	退款下载	编辑 删除
*46120047407106	wsphxd	微信视频号小店	[店铺名称]	1713746119921577986	1712309858862182402	订单更新	编辑 删除

共 192 条数据 < 1 / 10 > 20条/页

5.7.5. 消息推送日志

消息推送日志中，展示了消息推送的记录，及推送的成功状态。

店铺名称：Adaptor 中接入时候选择的店铺名称

平台：店铺所属的平台名称

目标服务名称：Adaptor 中接入时候填写的公司名称，Adaptor SaaS 一般取当前登录的租户名称，Adaptor 套装一般取喷码信息；支持自定义维护

推送 topic：Adaptor 中接入时候选择的消息类型

推送状态：消息推送的状态：（未推送：消息刚刚进入还没有触发推送；推送中：消息服务正在把订阅消息推送给订阅者中；推送成功：消息推送订阅者成功；推送失败：消息推送订阅者失败；停止推送：消息推送订阅者失败达到一定数量之后不再继续推送）

重新推送操作：重新触发消息推送给订阅者

查看异常：消息推送失败的，可查看失败的原因

查看报文：可查看消息推送的消息体

Adaptor 我的工作台 店铺对照管理 推送消息 字典表管理 目标服务信息 店铺与信息推送topic关系 admin

目标服务名称: 选择目标信息服务 平台: 选择平台 推送失败预警: 选择推送失败预警 推送状态: 选择推送状态 查询 重置 全部条件

批量重新推送 刷新

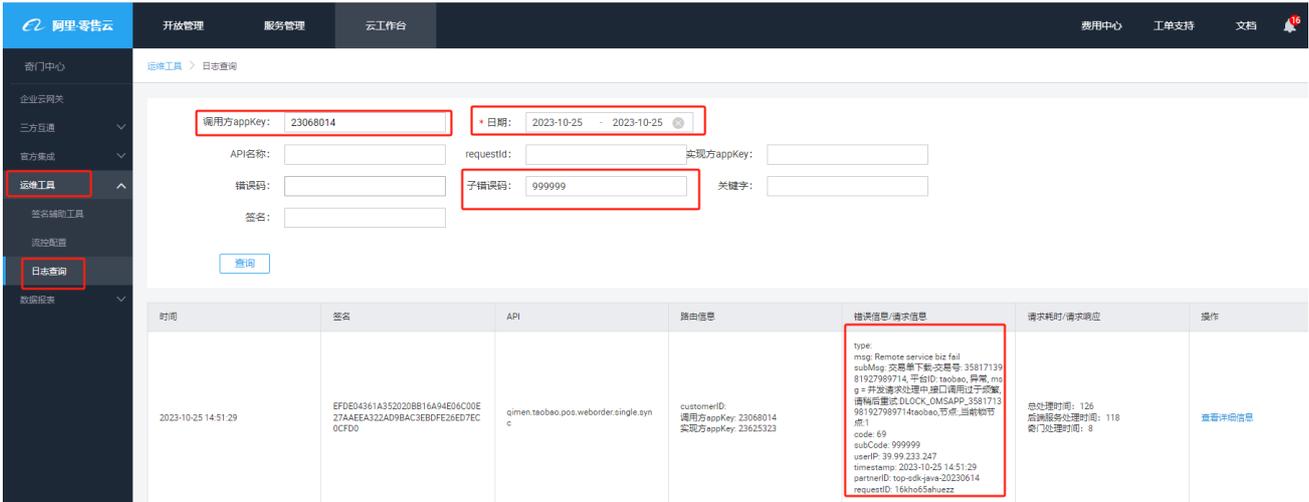
id	平台	推送失败预警	推送状态	推送的topic	店铺名称	店铺id	操作
1717427937451520002	微信视频号小店	正常	推送成功	订单更新	...	1665848185075081218	重新推送 查看详情 查看报文
1717427881314955265	微信视频号小店	正常	推送失败	订单更新	...	1694902364964741122	重新推送 查看详情 查看报文
1717427921290866690	微信视频号小店	正常	推送成功	退款下载	...	1665848185075081218	重新推送 查看详情 查看报文
1717427899455320066	微信视频号小店	正常	推送成功	退款下载	...	1665848185075081218	重新推送 查看详情 查看报文
1717427898926837761	微信视频号小店	正常	推送成功	退款下载	...	1665848185075081218	重新推送 查看详情 查看报文
1717427877791739905	微信视频号小店	正常	推送成功	订单更新	...	1665848185075081218	重新推送 查看详情 查看报文
1717427868039983106	微信视频号小店	正常	推送成功	订单更新	...	1665848185075081218	重新推送 查看详情 查看报文
1717427857055100929	微信视频号小店	正常	推送成功	订单更新	...	167869225248997889	重新推送 查看详情 查看报文
1717427856920883201	微信视频号小店	正常	推送成功	订单更新	...	1665848185075081218	重新推送 查看详情 查看报文
1717427847454339073	微信视频号小店	正常	推送成功	订单更新	...	1665848185075081218	重新推送 查看详情 查看报文
1717427847022325762	微信视频号小店	正常	推送成功	订单更新	...	1665848185075081218	重新推送 查看详情 查看报文
1717427844648349698	微信视频号小店	正常	推送成功	订单更新	...	1665848185075081218	重新推送 查看详情 查看报文
1717427841301295105	微信视频号小店	正常	推送成功	订单更新	...	1665848185075081218	重新推送 查看详情 查看报文
1717427833302757378	微信视频号小店	正常	推送成功	退款下载	...	1665848185075081218	重新推送 查看详情 查看报文

共 5636706 条数据 < 1 / 281836 > 20条/页

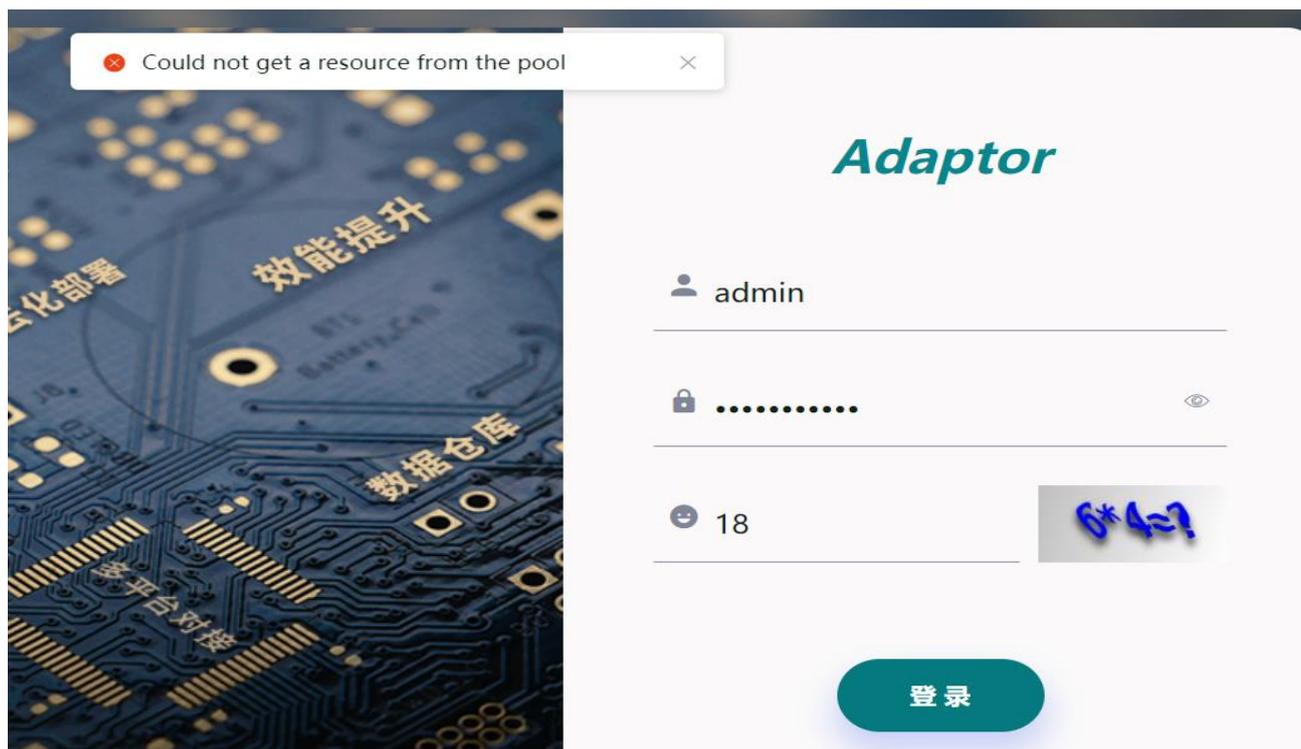
六、常见问题集

6.1. 订单通

- 1、中台创建店铺的时候报错：外部系统对接配置未找到启用配置
 - a) 排查外部系统参数配置，并且是启用状态的
 - b) 排查中台调用 adaptor 的地址是不是对应的环境，adpator 有公有云、聚石塔、云鼎等环境
- 2、订单下载下来，推送中台失败
 - a) 标准订单界面：手动点击推送重试，如果是推送提示“网络响应超时,请稍后重试”，请检查“外部系统参数配置”的 url 是否是对的，外部是否可以访问。
 - b) 查看推送失败的错误原因，查看下游 oms 返回具体错误信息，联系下游 oms 人员确认 oms 服务是否可用。
 - c) 如果是走奇门接口的，可以登录奇门后台，到【运维工具】-【日志查询】中查看对应的错误信息（登录地址：<https://qimen.taobao.com/>）



- 3、 订单下载下来，推送成功，中台找不到
 - a) Adaptor 如果显示推送成功，那么表示调用下游 oms 返回成功了，具体需要找下游 oms 人员进行排查
 - b) 如果是走奇门接口的，可以登录奇门后台，到【运维工具】-【日志查询】中查看对应的错误信息（登录地址：<https://qimen.taobao.com/>）
- 4、 订单未下载排查
 - a) 检查下商店管理-店铺的拉单时间，看看是否拉单时间是正常的
 - b) 检查下平台单据是否都自动格式化
- 5、 订单未自动格式化
 - a) 过滤未格式化的数据，检查最后几单的数据，看看是否时间比较久的
 - b) 手动格式化最后几单未格式化的数据，如果报错，检查下字段，是否存在超长的情况，主要查看金额字段
- 6、 中台单据的字段缺失
 - a) 标准订单界面：查看报文。看下是否有正确返回，如果有缺少请联系 adaptor 人员。如果没有缺少，请联系下游 oms 人员。
- 7、 发货回写调用 adaptor 接口报错
 - a) 查看 adaptor 日志管理-回写平台接口日志，上面有中台请求和响应报文以及平台的请求和响应报文，根据平台返回的数据分析参数是否有误
- 8、 抖音、拼多多、淘宝、京东订单手动下载报订单不存在
 - a) 检查下店铺参数是否有误，是否填写了不正确的店铺参数
 - b) 如果 a) 没有异常的，需要联系服务中心检查下是否平台的 rds 没有开启推送或者当前单据未推送
- 9、 淘宝订单推送报错：qimen route error | The specified route could not be found
 - a) 联系服务中心相关人员，在奇门中配置路由信息
- 10、 淘宝订单推送报错：没有对目前 APP KEY 进行授权
 - a) 检查下是否 POSKEY 未订购，或者订购已过期，请参考 4.1.2.6. 奇门 POS 配置章节
- 11、 店铺 token 未自动刷新问题
 - a) 检查下 appkey 管理列表，看是否有 token 过期时间字段，adaptor 通过 token 过期时间来判断是否需要 token 刷新
 - b) 微信视频号小店的，如果重新设置过 secret 的话，那么也会导致 token 无法刷新
 - c) 微信视频号小店等有白名单的，如果没有添加 adaptor 服务器到白名单中，也会导致 token 刷新失败
- 12、 系统启动正常，但是登录时提示拿不到线程池问题
 - a) 白名单是否有配置，看日志有没提示白名单问题
 - b) redis 配置是否正确，看 nacos 的 application.yaml 的两个 redis 配置是否一致



```

25 #时间+8小时
26 jackson:
27   date-format: yyyy-MM-dd HH:mm:ss
28   time-zone: GMT+8 #或UTC
29 redis:
30   host: r-1[redacted].redis.zhangbei.rds.aliyuncs.com
31   port: [redacted]
32   # 密码 没有则可以不用填
33   password: [redacted]
34   # 如果使用的jedis 则将lettuce改成jedis即可
35   lettuce:
36     pool:
37       # 最大活跃链接数 默认8
38       max-active: 100
39       # 最大空闲连接数 默认8
40       max-idle: 100
41       # 最小空闲连接数 默认0
42       min-idle: 1
43   messages:
44     basename: i18n/messages
45     encoding: UTF-8
46     use-code-as-default-message: true
47
48
49 mybatis-plus:
50   mapper-locations: classpath:mapper/*Mapper.xml
51   configuration:
52     #配置控制台打印完整带参数SQL语句
53     # log-impl: org.apache.ibatis.logging.stdout.StdOutImpl
54     #下划线转驼峰
55     map-underscore-to-camel-case: true
56 ---
57 redis:
58   hosts: r-k[redacted].redis.zhangbei.rds.aliyuncs.com
59   ports: [redacted]
60   password: [redacted]
61   maxIdle: 100
62   maxTotal: 100
63   timeout: 300000
64
65 logging:

```

- 13、 淘宝的消息推送到 rocketmq 时提示 topic 问题
 - a) 登录 rocketmq 配置后台看 GROUP 和 topic 等是否有配置
 - b) 排查配置的前缀是否和 nacos 配置的 profile 前缀一致

```

2023/10/14 15:15:50 at java.lang.Thread.run(Thread.java:750)
2023/10/14 15:15:50 Caused by: com.aliyun.openservices.shade.com.alibaba.rocketmq.client.exception.MQClientException: No route info of this topic,
MQ_INST_1993218193147477_BYt5eZ8Jw adaptor_queue_item
2023/10/14 15:15:50 See http://rocketmq.apache.org/docs/faq/ for further details.
2023/10/14 15:15:50 at com.aliyun.openservices.shade.com.alibaba.rocketmq.client.impl.producer.DefaultMQProducerImpl.sendDefaultImpl(DefaultMQProducerImpl.java:585)
2023/10/14 15:15:50 at com.aliyun.openservices.shade.com.alibaba.rocketmq.client.impl.producer.DefaultMQProducerImpl.send(DefaultMQProducerImpl.java:1136)
2023/10/14 15:15:50 at com.aliyun.openservices.shade.com.alibaba.rocketmq.client.impl.producer.DefaultMQProducerImpl.send(DefaultMQProducerImpl.java:1090)
2023/10/14 15:15:50 at com.aliyun.openservices.shade.com.alibaba.rocketmq.client.producer.DefaultMQProducer.send(DefaultMQProducer.java:257)
2023/10/14 15:15:50 at com.aliyun.openservices.ons.api.impl.rocketmq.ProducerImpl.send(ProducerImpl.java:123)
2023/10/14 15:15:50 ... 158 common frames omitted
2023/10/14 15:15:50 2023-10-14 15:15:50.628 [qtp1660536720-32] INFO c.b.e.a.service.rabbitmq.producer.PushToE3Producer.send[PushToE3Producer.java:172] - 开始发送消息,
exchangeName: adaptor_exchange_item, queueName: item_routingName: adaptor_routing_item, message

```

- 14、 平台拉单时间一直没变
 - a) 排查店铺的 token 是否过期
 - b) 排查日志看是否请求频率超了或者请求间隔时间超过平台限制
- 15、 拉平台详情失败，数据堆积在单据队列表中
 - a) 排查店铺的 token 是否过期
 - b) 看是否存在存在堆积过多查不到数据的历史订单

- c) 通过日志排查请求是否超频了
- 16、 单据一直未格式化
 - a) 看格式化调度是否有开启
 - b) 手动格式化判断是否平台数据问题导致格式化失败
 - c) 通过日志排查是否特殊字段处理有问题
- 17、 服务启动提示 liquebase 异常
 - a) 联系开发人员调整数据库脚本异常的脚本和 id
- 18、 中台回写失败
 - a) 通过回写平台接口日志中记录的日志排查是系统问题还是平台问题
- 19、 订单格式化后的标准地址异常
 - a) 看平台订单的省市区地址是否都有且正确
 - b) 通过平台行政区域映射看是否配置有问题
 - c) 看标准行政区域地址信息是否正确
- 20、 发票数据没有走单独的发票推送接口
 - a) 参数配置中有个订单发票是否推送开关打开
- 21、 淘宝，京东等 rds 拉不到单据
 - a) 排查店铺配置的 source_rds 是否正确
 - b) 店铺的 seller_nick 等是否正确，是否重复配置
 - c) 单据是否在 rds 数据时限内，比如淘宝的 rds 只保留最近 3 个月的数据

6.2. 打印通

1、 模板说明

取号时没有传 stdTemplateName 指定模板时，系统默认随机一个模板，优先会使用含有自定义区 url 的模板

- a) 如果要指定模板可以传 stdTemplateName 字段，该字段的值需要传存在的平台模板名称
- b) 如果同一个标准模板下配置了多个自定义区模板，则除了指定 stdTemplateName 还需要传 customAreaName 字段，该字段的值需要传存在的自定义模板名称

平台模板名称	模板Url	自定义模板名称	自定义模板Url	平台	店铺
顺丰-无界(76x130)	https://template-content.j...			京东	京东
顺丰-无界(76x130双条码)	https://template-content.j...			京东	京东
顺丰-无界(100x180)	https://template-content.j...			京东	京东
顺丰-无界(新_100x150)	https://template-content.j...	1	https://storage.jd.com/jdl-...	京东	京东
顺丰-无界(新_100x150)	https://template-content.j...	SF2_在用	https://storage.jd.com/jdl-...	京东	京东
顺丰-无界(新_100x180)	https://template-content.j...	顺丰无界新100x180	https://storage.jd.com/jdl-...	京东	京东
顺丰-无界(新_100x180)	https://template-content.j...	顺丰无界_100x180_05_15	https://storage.jd.com/jdl-...	京东	京东
顺丰-无界(新_100*210)	https://template-content.j...			京东	京东
无界-顺丰-含自定义 (76x...	https://template-content.j...			京东	京东

2、如何使用面单自定义区

a) 在平台后台配置模板的自定义区变量

变量格式(其中 key 为自己定义的变量名):

淘宝、抖音、快手: <%=_data.key%>

拼多多: <%=data.key%>

京东: @{key}

华为商城: <%=key%>

b) 调用获取面单打印文件接口传入需要打印的数据:

例如 custom_data 传 "{ \"key\": \"我是打印内容\" }"

3、门店发货对总怎么取号

isSumUp 传 true, deliveryType 传 store

a) 淘宝

i. 物流如果是直营物流(如顺丰、德邦)则需要传 settleAccount 月结卡号字段

ii. 物流如果是加盟物流则需要传 branchCode 网点编码(目前淘宝仅支持中通)

b) 抖音

i. 物流如果是直营物流(如顺丰、德邦)则需要传 settleAccount 月结卡号字段

ii. 物流如果是加盟物流: 传 branchCode、settleAccount、secretKey、whCode(具体看该物流是否有相关字段的概念)

iii. 具体支持物流参考平台文档:
<https://bytedance.feishu.cn/docs/doccnIdKDVpdz3tZWYSgRfQFPyh#QG7kVe>

c) 快手

i. 物流如果是直营物流(如顺丰、德邦等)则需要传 settleAccount 月结卡号字段

ii. 物流如果是加盟物流: 传 branchCode、settleAccount、secretKey、whCode(具体看该物流是否有相关字段的概念)

d) 拼多多

i. 平台不支持总对总

e) 京东

i. 物流如果是直营物流(如顺丰、德邦)则需要传 settleAccount 月结卡号字段

ii. 加盟不支持

4、线下或小平台订单如何获取电子面单

需要依赖具有电子面单能力的平台去取号,

比如微盟的单子可以通过淘宝平台代理获取电子面单:

则 platform_code=taobao, orderChannel=WEI_MENG

比如线下单子可以通过抖音平台代理获取电子面单:

则 platform_code=douyinxiaodian, orderChannel=OTHERS

具体传值查看字段描述

5、拼多多如何直连打印机

获取面单文件的类型为 3(接口 4.2 preViewType=3), 拿到打印结果(perPrintPDF)后, 需要与拼多多打印组件进行交互。

直接将 perPrintPDF 发给打印组件, 参考如下文档---发送打印数据协议(print):

<https://open.pinduoduo.com/application/document/browse?idStr=3BBB4C229B6A8FCC>

4. 发送打印数据协议(print)

请求协议格式 (加密) 如下

```
Plain Text 收起 ^
{
  "ERPIId": 303379400,
  "cmd": "print",
  "requestID": "90E06437",
  "task": {
    "documents": [
      {
        "contents": [
          {
            "addData": {
              "sender": {
                "address": {
                  "city": "上海市",
                  "detail": "陕西面馆",
                  "district": "长宁区",
                  "province": "上海市"
                },
                "mobile": "333333333333",
                "name": "ceshi"
              }
            }
          }
        ]
      }
    ]
  }
}
```

6. 常见报错信息

a) 淘宝

- i. API 字段映射错误, 请提供参数信息联系小二处理。字段名: oaid, 非淘系订单不用传 oaid。
- ii. 商家请求参数错误, 固话和手机只能包含+、-、空格、单引号、数字、英文逗号、中文逗号
如果是加密或脱敏的手机号需要传 oaid 字段。
- iii. [taobao decrypt result is null]: 淘宝解密收件人返回内容为空
传的 oaid 或 sourceBillNo 平台订单号值不对

b) 抖音

- i. 商家未开通电子面单服务
报错的原因是取号接口时传的发件人地址和平台开通的地址不一致
- ii. 业务异常 || 详细地址长度不能超过 200 字符, 请检查重试
地址密文错误, 导致平台认为是明文, 触发了明文的地址校验。

c) 拼多多

- i. 业务服务错误 | 详细地址字段长度超过限制
地址密文错误, 导致平台认为是明文, 触发了明文的地址校验。

d) 京东

- i. 详细地址存在冲突执行全地址匹配, 发送人地址未匹配商家审核成功的签约信息地址
是因为平台开通的签约地址存在多条相同的, 程序无法匹配使用哪一个地址下的月结卡号/网点编号,

需要指定月结卡号 settleAccount（直营）或网点编号 branchCode（加盟）

7、其他

- 京东取号地址只支持明文不支持密文。手机号和电话即支持明文也支持密文。其他平台地址和手机号都支持密文
- 取号或打印时给的密文要完整且不能混入明文，注意是否有数据库字段限制导致密文被截断。
- 同一个平台订单多次取号会返回相同的运单号，需要传不一样的包裹号才能获取新的运单号，详见下单取号接口的 electronicTakeNo 字段说明
- 非对总取号时，发货的省市区详细地址要与平台开通的地址一致，可在如下 adaptor 界面查看



七、附录

持续更新请关注：<http://bsconfluence.baison.net:40506/x/dTXeC>

7.1. SDK 下载



Sdk 下载：`sdk-java-1.0.4.jar`

下载 sdk 的时候，请选择上面的 sdk 进行复制，再黏贴到本地的方式进行下载。

7.2. Adaptor 全网通配置文件加解密工具下载



adaptor 全网通配置文件加解密工具下载：`adaptor全网通配置文件加解密工具.jar`

下载工具的时候，请选择上面的工具进行复制，再黏贴到本地的方式进行下载，下载之后双击即可运行。